(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平11－242676

(43)公開日　平成11年(1999)９月７日

(51)Int.Cl.[6]	識別記号	ＦＩ		
Ｇ０６Ｆ　17/30		Ｇ０６Ｆ	15/40	３７０Ａ
				３４０
			15/403	３１０Ａ

審査請求　未請求　請求項の数20　ＯＬ　（全 89 頁）

(21)出願番号　特願平10－43187

(22)出願日　平成10年(1998)２月25日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　岡本　卓哉
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所情報通信開発本部内

(72)発明者　高橋　亨
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所情報通信開発本部内

(72)発明者　川口　久光
東京都千代田区神田駿河台四丁目６番地
株式会社日立製作所内

(74)代理人　弁理士　小川　勝男

最終頁に続く

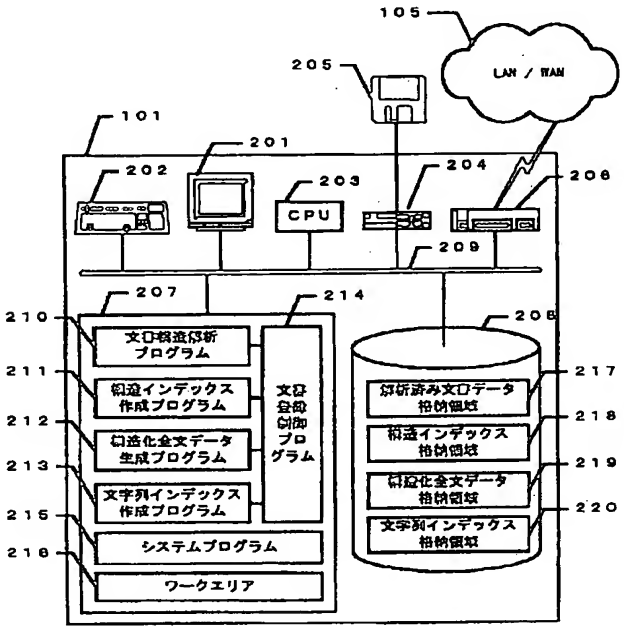
(54)【発明の名称】　構造化文書登録方法、検索方法、およびそれに用いられる可搬型媒体

(57)【要約】

【課題】構造化文書を対象とした構造指定検索において、論理要素の文書中における出現位置に関する条件を指定できるようにして精度の高い構造指定検索を可能にすること。

【解決手段】文書をデータベースに登録する際、各登録対象文書の持つ論理構造を重ね合わせ、文書中での出現位置が等しい構造要素を単一のメタノードによって代表させた構造インデックスを作成し、文書検索時には構造インデックスを参照して指定された構造条件を満足するメタノードの集合を求め、それらのメタノードの識別子をキーとして文字列インデックスを検索することにより、指定条件を満たす文書群を求める。これにより、構造化文書の集合からなる文書データベース上において、精度の高い構造指定検索が可能となる。

図２



1

【特許請求の範囲】

【請求項１】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

登録対象文書の持つ論理構造を解析して得られる解析済み文書データを文書データベースに登録する解析済み文書データ生成登録ステップと、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップを有することを特徴とした構造化文書の登録方法。

【請求項２】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

各登録対象文書について、該文書中に含まれる各文字列データから、所定の部分文字列と、該部分文字列の登録対象文書中における文字位置情報と、前記登録対象文書を文書データベース中で一意に識別する文書識別子と、前記部分文字列を含む文字列データを前記構造インデックス中で代表するメタノードの文脈識別子とを抽出し、前記文字位置情報および前記文書識別子ならびに前記文脈識別子からなる構造化文字位置情報を生成し、前記部分文字列と前記構造化文字位置情報との対応関係を登録して文字列インデックスを更新する文字列インデックス更新ステップを有することを特徴とした構造化文書の登録方法。

【請求項３】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップにおいて、二つの登録対象文書の持つ木構造を重ね合わせる際に、

両文書の木構造を構成する二つのノードを比較したときに、両ノードが共に木構造の根であれば両ノードは互いに対応していると見なし、

両ノードが根でない場合には、両ノードの上位ノードが互いに対応し、かつ両ノードの種別が一致し、かつ種別の等しい兄弟ノードの並びの先頭から正順に数えた出現順序が等しい場合に両ノードは対応していると見なし、

対応している両ノードを単一のメタノードによって代表するように前記構造インデックスを構築することを特徴とした構造化文書の登録方法。

2

【請求項４】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップにおいて、二つの登録対象文書の持つ木構造を重ね合わせる際に、

両文書の木構造を構成する二つのノードを比較したときに、両ノードが共に木構造の根であれば両ノードは互いに対応していると見なし、

両ノードが根でない場合には、両ノードの上位ノードが互いに対応し、かつ両ノードの種別が一致し、かつ種別の等しい兄弟ノードの並びの末尾から逆順に数えた出現順序が等しい場合に両ノードは対応していると見なし、

対応している両ノードを単一のメタノードによって代表するように前記構造インデックスを構築することを特徴とした構造化文書の登録方法。

【請求項５】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップにおいて、二つの登録対象文書の持つ木構造を重ね合わせる際に、

両文書の木構造を構成する二つのノードを比較したときに、両ノードが共に木構造の根であれば両ノードは互いに対応していると見なし、

両ノードが根でない場合には、両ノードの上位ノードが互いに対応し、かつ両ノードの種別が一致し、かつ種別の等しい兄弟ノードの並び中において、両ノードが共に該並びの先頭位置にあるか、または両ノードが共に先頭以外の位置にある場合に、両ノードは対応していると見なし、

対応している両ノードを単一のメタノードによって代表するように前記構造インデックスを構築することを特徴とした構造化文書の登録方法。

【請求項６】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

各登録対象文書の持つ論理構造を順次重ね合わせ、文書中における出現位置および種別が同一である構造要素群を単一のメタノードによって代表させた構造インデックスを作成し、該構造インデックスの木構造を構成する各メタノードに、該メタノードを一意に識別する文脈識別子を与える構造インデックス作成ステップにおいて、二つの登録対象文書の持つ木構造を重ね合わせる際に、

両文書の木構造を構成する二つのノードを比較したときに、両ノードが共に木構造の根であれば両ノードは互いに対応していると見なし、

両ノードが根でない場合には、両ノードの上位ノードが互いに対応し、かつ両ノードの種別が一致し、かつ種別の等しい兄弟ノードの並び中において、両ノードが共に該並びの末尾位置にあるか、または両ノードが共に末尾以外の位置にある場合に、両ノードは対応していると見なし、

対応している両ノードを単一のメタノードによって代表するように前記構造インデックスを構築することを特徴とした構造化文書の登録方法。

【請求項７】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理の過程で、

登録対象文書の持つ論理構造を解析して得られる解析済み文書データを文書データベースに登録する際に、

該解析済み文書データ中から、検索対象としては不適切な構造および内容文字列を抽出し、該構造および内容文字列を解析済み文書データ中から削除した後に、

該解析済み文書データを文書データベースに登録するステップを有することを特徴とした構造化文書の登録方法。

【請求項８】請求項１に記載の構造化文書の登録方法において、

前記解析済み文書データ生成登録ステップおよび前記構造インデックス作成ステップを実行することにより作成された構造インデックスを記録した可搬型媒体。

【請求項９】請求項１に記載の構造化文書の登録方法において、

前記解析済み文書データ生成登録ステップおよび前記構造インデックス作成ステップを実行する機能を備えたコンピュータプログラムを記録した可搬型媒体。

【請求項１０】請求項２に記載の構造化文書の登録方法において、

前記文字列インデックス更新ステップを実行することにより作成された前記文字列インデックスを記録した可搬型媒体。

【請求項１１】請求項２に記載の構造化文書の登録方法において、

前記文字列インデックス更新ステップを実行する機能を備えたコンピュータプログラムを記録した可搬型媒体。

【請求項１２】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の検索方法であって、

文書の検索を行う処理が、

構造インデックスを参照し、指定された構造条件を満たす文脈識別子の集合を決定する構造条件判定ステップと、

検索タームから所定の部分文字列を抽出し、文字列インデックスを参照して該部分文字列に対応する構造化文字位置情報の集合を抽出する構造化文字位置情報抽出ステップと、

前記構造化文字位置情報の集合中から、前記構造条件判定ステップで決定した集合中に含まれる文脈識別子をもち、かつ前記検索ターム上における部分文字列の並びと同じ位置関係をもつ構造化文字位置情報を抽出するインデックス検索ステップを有することを特徴とした構造化文書の検索方法。

【請求項１３】請求項１２に記載の構造化文書の検索方法において、

前記構造条件判定ステップおよび前記構造化文字位置情報抽出ステップおよび前記インデックス検索ステップを実行する機能を備えたコンピュータプログラムを記録した可搬型媒体。

【請求項１４】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

登録される文書の持つ論理構造を解析して得られる解析済み文書を作成するステップと、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルの内容を解析して、同じ種別とみなされる構造を得るステップと、

最上位の構造が同じ種別であるとみなされる、複数の解析済み文書の論理構造を重ね合わせ、２つ以上の文書において同じ位置、種別とみなされる構造を１つのメタノードに代表させ、１つの文書のみに出現するとみなされる構造は、その構造を表わす１つのメタノードを生成することで、最上位の構造が同じとみなされる複数の登録文書ごとに、共通の構造情報をメタノードを要素とする木構造で表わす構造インデクスを作成するステップと、

最上位の構造の種別が同じとみなされる複数の登録文書ごとに作成された複数の構造インデクスの最上位構造を表わすメタノードの共通の上位構造として、登録文書に対応する構造を持たないルートメタノードを作成し、各登録文書の最上位構造を表わすメタノードは、ルートメタノードの子構造とした、ルートメタノードを最上位ノードとするメタ構造インデクスを作成するステップと、

ルートメタノードを含めてメタ構造インデクスを構成するメタノードを一意に識別する文脈識別子を各構造インデクスを構成するメタノードに与えるステップと、

5

各メタノードに対応した各登録文書の構造内の文字列を登録文書の識別情報と文脈識別子の情報と共に文字列インデクスに登録するステップを有することを特徴とした構造化文書の登録方法

【請求項１５】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

登録される文書の持つ論理構造を解析して得られる解析済み文書を作成するステップと、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルの内容を解析して、同じ種別とみなされる構造を得るステップと、

最上位の構造が同じ種別であるとみなされる、複数の解析済み文書の論理構造を重ね合わせ、２つ以上の文書において同じ位置、種別とみなされる構造を１つのメタノードに代表させ、１つの文書のみに出現するとみなされる構造は、その構造を表わす１つのメタノードを生成することで、最上位の構造が同じとみなされる複数の登録文書ごとに、共通の構造情報をメタノードを要素とする木構造で表わす構造インデクスを作成するステップと、

最上位の構造の種別が同じとみなされる複数の登録文書ごとに作成された複数の構造インデクスの最上位構造を表わすメタノードの共通の上位構造として、登録文書に対応する構造を持たないルートメタノードを作成し、各登録文書の最上位構造を表わすメタノードは、ルートメタノードの子構造とした、ルートメタノードを最上位ノードとするメタ構造インデクスを作成するステップと、

ルートメタノードを含めてメタ構造インデクスを構成するメタノードを一意に識別する文脈識別子を各構造インデクスを構成するメタノードに与えるステップと、

各メタノードに対応した各登録文書の構造内の文字列を登録文書の識別情報と文脈識別子の情報と共に文字列インデクスに登録するステップと、

同じ種別を持つ、異なる位置に出現する構造に対する共通名称を別名として設定し、該別名とメタノードの文脈識別子を対応付けた、別名構造インデクスを生成するステップを有することを特徴とした構造化文書の登録方法

【請求項１６】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

登録される文書の持つ論理構造を解析して得られる解析済み文書を作成するステップと、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルの内容を解析して、同じ種別とみなされる構造を得るステップと、

最上位の構造が同じ種別であるとみなされる、複数の解析済み文書の論理構造を重ね合わせ、２つ以上の文書において同じ位置、種別とみなされる構造を１つのメタノードに代表させ、１つの文書のみに出現するとみなされる構造は、その構造を表わす１つのメタノードを生成することで、最上位の構造が同じとみなされる複数の登録文書ごとに、共通の構造情報をメタノードを要素とする木構造で表わす構造インデクスを作成するステップと、

各構造インデクスを構成するメタノードを一意に識別する文脈識別子を構造インデクスを構成するメタノードに与えるステップと、

各構造インデクスを識別するための構造インデクス識別子を構造インデクスに与えるステップと、

各メタノードに対応した各登録文書の構造内の文字列を登録文書の識別情報と構造インデクス識別子と文脈識別子の情報と共に文字列インデクスに登録するステップと、　複数の構造インデクスにまたがって同じ種別であるとみなされる特定の種別を持つ構造に対する共通名称を別名として設定した別名テーブル共通名称定義テーブルを読み出し、共通名称と各構造インデクスの構造インデクス識別子およびメタノードの文脈識別子を対応付けた、共通名称構造インデクスを生成するステップ同じ種別を持ち異なる構造インデクス、または位置に出現する構造に対する共通名称を別名として設定し、該別名と構造インデクス識別子およびメタノードの文脈識別子を対応付けた、別名構造インデクスを生成するステップを有することを特徴とした構造化文書の登録方法

【請求項１７】予め登録した文書の集合を対象として文書内容の検索を行う文書検索システムにおける構造化文書の登録方法であって、

文書の登録を行う処理が、

登録される文書の持つ論理構造を解析して得られる解析済み文書を作成するステップと、

文書の要素型名と構造の種別の対応を定義した種別定義テーブルの内容を解析して、同じ種別とみなされる構造を得るステップと、

各文書の最上位構造の上に全文書に共通な仮の最上位構造を追加した文書に変換するステップと、

該仮の最上位構造を最上位の構造として、複数の解析済み文書の論理構造を重ね合わせ、２つ以上の文書において同じ位置、種別とみなされる構造を１つのメタノードに代表させ、１つの文書のみに出現するとみなされる構造は、その構造を表わす１つのメタノードを生成することで、共通の構造情報をメタノードを要素とする木構造で表わす構造インデクスを作成するステップと、

構造インデクスを構成するメタノードを一意に識別する文脈識別子をメタノードに与えるステップと、

各メタノードに対応した各登録文書の構造内の文字列を登録文書の識別情報と文脈識別子の情報と共に文字列インデクスに登録するステップを有することを特徴とした構造化文書の登録方法

【請求項１８】請求項１２に記載の構造化文書の検索方法であって、

文書の検索を行う処理が、
構造インデックスを参照し、指定された構造条件を満たす文脈識別子の集合を決定する構造条件判定ステップにおいて、
文書の要素型名と構造の種別の対応を定義した種別定義テーブルを用いて、検索条件に記載された要素型名を種別情報に変換した上で、構造インデクスを参照して、種別情報に適合する文脈識別子の集合を決定することを特徴とした構造化文書の検索方法。

【請求項１９】請求項１２に記載の構造化文書の検索方法であって、
文書の検索を行う処理が、
構造インデックスを参照し、指定された構造条件を満たす文脈識別子の集合を決定する構造条件判定ステップにおいて、
共通名称インデクスを用いて検索条件に記載された要素型名を基に、構造インデクスの対応するノードの位置を取得することにより、種別情報に適合する文脈識別子の集合を決定することを特徴とした構造化文書の検索方法。

【請求項２０】請求項１に記載の構造化文書の検索方法であって、
文書の登録を行う処理において、登録文書の構造に付けられた名称を構造インデクスにおける構造の重ね合わせのために種別に変換するステップを有することを特徴とした構造化文書の登録方法

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、コンピュータ装置を用いた文書検索システムや文書管理システム等における文書登録方法および文書検索方法に係わり、特に、各文書がそれぞれ論理的な構造を備えている構造化文書の集合を対象として、特定の文書内容の検索を高速に行うことのできるようにした構造化文書の登録方法，検索方法、およびそれに用いられる可搬型媒体に関する。

【０００２】

【従来の技術】情報化社会の本格的な進展に伴い、ワードプロセッサ、パーソナルコンピュータ等を用いて作成された電子化文書情報が爆発的な勢いで増加しつつある。このような状況下において、蓄積された膨大な電子化文書群の中から、必要とする情報を含んだ文書を高速かつ確実に検索したいという要求が高まっている。

【０００３】このような要求に応える技術として全文検索がある。全文検索では登録対象文書中のテキスト全体を計算機システムに投入してデータベース化し、該データベース上で指定文字列（以後、検索タームと呼ぶ）を直接検索するので、キーワードを必要とせず、原理的に検出漏れのない検索が可能となる。

【０００４】また、例えばＳＧＭＬ（ISO 8879:1986, Standard Generalized Markup Language）で記述された文書など、文書を構成する個々の論理的な構造要素が識別できる文書（以下、構造化文書と呼ぶ）を対象として、論理構造に関する条件を検索条件中に付加した検索（以下、構造指定検索と呼ぶ）を行うことにより、精度の高い検索を実現することができる。

【０００５】構造指定検索を可能にする方式としては、例えば特開平８－１４７３１１号公報に示す発明（以下、公知例１と呼ぶ）で提案した検索方法がある。以下、公知例１の概要について説明する。

【０００６】公知例１の構造化文書検索方法は、文書を登録する際、まず登録文書をそのまま本文として検索用データベースに登録する。

【０００７】次に、登録した本文について各論理構造の先頭を表わす特定の文字列（これを公知例１では前方マーカと呼ぶ）および末尾を表わす特定の文字列（これを公知例１では後方マーカと呼ぶ）を検出することにより論理構造の識別処理を行うとともに、本文を論理構造ごとに分割する処理を行う。例えば、電子出願特許明細書の場合には、論理構造「要約」の範囲を示す前方マーカとして“<SDO ABJ>”を、後方マーカとして“</SDO>”を検出し、これらによって囲まれるテキストを「要約」に対応する本文として切り出す。他の論理構造についても同様な切り出しを行い、本文を論理構造ごとに分割する。

【０００８】次に、分割された各論理構造に対応する本文について、それぞれ凝縮本文の作成処理を行う。すなわち、「要約」については「要約」に関する本文テキストを単語単位に部分文字列に分割し、分割した部分文字列間で相互に包含関係を調べ、他の部分文字列に含まれる文字列を排除することにより「要約」に関する凝縮本文を作成する。他の論理構造についても同様な処理を行うことにより論理構造別の凝縮本文を作成し、これを凝縮本文ファイルとして検索用データベースに登録する。

【０００９】次に、本文中に現れた文字の文字コードに対応するビットに‘１’を設定することにより文字成分表を作成し、これを検索用データベースに文字成分表ファイルとして登録する。

【００１０】このようにして検索用データベースを構築した後、公知例１では次のようにして文書検索処理を行う。

【００１１】はじめに、指定された検索タームを文字単位に分解し、検索タームを構成するすべての文字がともに含まれる文書を上記文字成分表を参照して抽出する。

【００１２】次に、各論理構造に対応する凝縮本文ファイルのうち、検索対象として指定された論理構造に関する凝縮本文ファイルをサーチ対象として選択するとともに、その中で文字成分表サーチによって抽出された文書の凝縮本文だけを対象としてサーチすることにより、指定された論理構造中に指定された検索タームが含まれている文書を抽出する。指定された検索条件式に複数の検

索ターム間の本文中での位置関係が指定されていない場合にはここで検索処理を終了する。他方、そのような位置関係が指定されている場合には凝縮本文サーチの結果抽出された文書に対応する本文の内容を読み、指定された検索タームがすべて含まれ、かつ検索ターム間の位置関係について、指定された条件が満たされるもののみを抽出する。

【００１３】このようにして、公知例１の方法によれば、大規模なテキストデータベースに対する実用的な検索速度を保ち、かつ構造指定検索を可能にすることができる。

【００１４】

【発明が解決しようとする課題】前記公知例１に示す従来技術によって、一定の構造指定検索が可能となる。しかし、公知例１による構造指定検索では意図した構造条件に従った検索ができない場合がある。

【００１５】公知例１の方式では、登録対象文書の持つ構造を予め定めておいたいくつかの部分構造に切り分け、各部分構造ごとに凝縮本文ファイルを作成し、検索時には、部分構造の構造名と凝縮本文ファイルとの対応を定義したテーブルを参照して検索対象とすべき凝縮本文ファイルの集合を決定し、該集合中に含まれる凝縮本文ファイルだけを対象として検索を行うことにより構造指定検索を実現する。

【００１６】この方式では、テキストデータベースを構築する段階で、将来どのような構造条件を指定した検索が行われるかを想定し、そのような条件に対応した検索が可能となるように凝縮本文ファイルの切り分けを行うので、データベース構築時に想定していなかった構造条件を指定した検索は不可能となる。

【００１７】例えば、文書が“要約”と“本体”の二つの論理要素（以下、要素と呼ぶ）から構成され、“本体”はさらに任意個数の“節”の繰り返しからなり、各“節”は１個の“節題”と任意個数の“段落”からなる場合を考える。このような構造を持つ文書群からなるテキストデータベースを構築する際に、凝縮本文ファイルを“要約”に対応するものと“本体”に対応するものの二つに分けて作成したとすると、「節題中に文字列○○を含む文書群を求める」という条件を満たす構造指定検索は不可能となってしまう。

【００１８】もちろん、“本体”全体で１個の凝縮本文ファイルとする代わりに、これをさらに“節題”と“段落”とに分けて凝縮本文ファイルを作成しておけばこの条件に対応することはできる。しかし、仮にこのようにファイルを構成しても、例えば「最初の節の内部（節題または段落中）に文字列○○を含む文書群を求める」、あるいは「節中で最後の段落に文字列××を含む文書群を求める」というような構造条件には対応できない。このような順位指定付きの構造条件に対応しようとすると、予め節および段落の出現順位ごとに別々に凝縮本文

ファイルを用意しておくことが必要となるが、節および節内の段落はそれぞれ任意個数出現しうるため凝縮本文ファイルの数が極めて多数となるだけでなく、公知例１は出現順位に関する任意の指定を含んだ構造条件から細分化された凝縮本文ファイルの集合への対応付けを行う手段を備えていないので、実際にはこのような条件を満たす検索は不可能となる。

【００１９】このように、前記従来技術では、文書中に現れた論理要素の出現位置に関する条件を構造条件指定中に含めることができないため、精度の高い構造指定検索ができないという問題があった。

【００２０】本発明の目的は、前記従来技術の持つ上記問題点を解決し、高精度でかつ効率のよい構造指定検索を行う機能を提供することにある。

【００２１】さらに、前記従来技術では、あらかじめ定められた１つの構造を持った文書群に対する構造指定検索しか実現できない。

【００２２】つまり、SGMLなどの構造化文書は、文書型定義（DTD: Document Type Definition）によりあらかじめ定められた構造を持つ文書である。したがって、特定の文書型定義に従った文書群に対する構造指定検索を行う場合は、出現しうる構造指定条件すべてに対応するように、文書を構造毎に切り分けることで、構造指定検索が可能となる。

【００２３】しかし、文書型定義は1つだけではない。例えば、論文、報告書などはそれぞれ異なる文書型定義を持っている。このように、構造化文書では、その文書の目的に応じて様々な文書構造を持っており、その文書構造に合わせた文書型定義が作成される。

【００２４】これらの文書を文書型定義毎にグループ化して登録することで、個々のグループにおいては、構造指定検索が可能となるが、全グループに共通して出現しうる構造を指定した検索を実現しようとすると、全グループに対して個々に構造指定検索を行ない、結果を統合しなければ、結果を得ることができない。

【００２５】また、XML（Extended Markup Language）のように、必ずしも特定の構造を持たなくても良い構造化文書の規格化がW3C（World Wide Web Consortium）で進められており、ＳＧＭＬのように特定のＤＴＤに従った文書構造を持つ文書だけが検索対象とならないことが考えられる。

【００２６】さらに、前記従来技術では、「タイトル」、「題目」のように、同じ意味（種別）を持つ構造であっても、要素型名が異なっていれば別の構造とみなされるため、「"タイトル"に"SGML"が含まれる文書」という構造指定検索では、「"題目"に"SGML"が含まれる文書」という条件を満たす文書は、検索結果として得ることができない。

【００２７】特に、文書型定義が異なる場合、それぞれの文書型定義毎に、同じ種別の構造に異なる要素型名が

付けられることが考えられる。

【００２８】例えば、「タイトル」に関する構造指定検索を行おうとすると、ユーザが「タイトル」、「題目」、「表題」、「TITLE」など、各文書型定義に出現する様々な「タイトル」を意味する要素型名を指定して構造指定検索条件を作成しなければ、必要とする文書をすべて取得することができない。また、登録文書の文書型定義をすべて知らなければ、ユーザが想定した要素型名で「タイトル」を意味する構造全てを網羅することはできない。例えば「T」という構造にタイトルを記述するといった文書型定義に従った文書は、それを知らないユーザからの構造指定検索では、決して取得することはできないであろう。

【００２９】本発明のもう一つの目的は、上記問題点を解決し、文書構造が異なる文書が混在する文書群を、高精度でかつ効率的に構造指定検索する機能を提供することである。

【００３０】さらに構造指定検索条件として、「章、節などのいづれかの項目のタイトルにおいて、"ＳＧＭＬ"という単語を含む文書」を設定した場合、"タイトル"という構造条件を満たす構造を全て探さなければならないため、検索効率が悪くなるという問題がある。

【００３１】検索条件として、「/文書/章/タイトル」のように、タイトルまでの全ての構造を最上位から順に指定するようにすれば、効率的に構造を特定することができる。しかし、ユーザが「"/文書/章/タイトル"または"/文書/章/節/タイトル"または、、、」のように、全ての構造を示した構造指定検索条件を作成しなければならないため、ユーザの負担が大きくなる上に、ユーザが検索対象の文書の構造を全ての把握しておかなければ、検索漏れが発生することになる。

【００３２】本発明のさらにもう一つの目的は、上記問題点を解決し、複数の階層に出現する同じ種別の構造を指定した検索を、複雑な構造条件を指定することなく、効率的に実現する機能を提供することである。

【００３３】

【課題を解決するための手段】上記課題を解決するため、本発明による文書登録および検索方法は、以下に示すステップを備える。

【００３４】すなわち、本発明による文書登録方法において、文書の登録処理は、（１）登録対象文書の持つ論理構造を解析して解析済み文書データを生成し、これを文書データベースに登録する解析済み文書データ生成登録ステップと、（２）各登録対象文書の持つ論理構造を、登録順に従って順次重ね合わせ、文書中における出現位置および種別が同じである要素群は単一のメタ要素によって代表させ、文書中における出現位置が同じである文字列データ群は単一のメタ文字列データによって代表させることにより、メタ要素群およびメタ文字列データ群（以下、これらを総称してメタノードと呼ぶ）の木

構造から構成される構造インデックスを生成し、該構造インデックスを構成するすべてのメタノードに対して、それらを構造インデックス中で一意に識別する識別子である文脈識別子を与える構造インデックス生成ステップと、（３）各登録対象文書について、その文書に対応する解析済み文書データ中に含まれるすべての文字列データと、その文字列データを構造インデックス中で代表するメタ文字列データの文脈識別子との対応関係の定義から構成される構造化全文データを生成する構造化全文データ生成ステップと、（４）各登録対象文書に対応する構造化全文データから、所定の部分文字列と、該部分文字列の前記登録対象文書中における文字位置情報と、前記登録対象文書を文書データベース中で一意に識別する文書識別子と、前記部分文字列を含む文字列データを構造インデックス中で代表するメタ文字列データの文脈識別子とを抽出し、前記文字位置情報および前記文書識別子および前記文脈識別子からなる構造化文字位置情報を生成し、前記部分文字列と前記構造化文字位置情報との対応関係を登録して文字列インデックスを更新する文字列インデックス更新ステップを有する。

【００３５】また、本発明による文書検索方法において、登録済み文書の検索処理は、（１）前記構造インデックスを参照し、指定された構造条件を満たす文脈識別子の集合を決定する構造条件判定ステップと、（２）検索タームから所定の部分文字列を抽出し、前記文字列インデックスを参照して該部分文字列に対応する構造化文字位置情報の集合を抽出する構造化文字位置情報抽出ステップと、（３）前記構造化文字位置情報の集合中から、前記構造条件判定ステップで決定した集合中に含まれる文脈識別子をもち、かつ前記検索ターム上における部分文字列の並びと同じ位置関係を持つ構造化文字位置情報を抽出するインデックス検索ステップを有する。

【００３６】さらに、本発明による文書検索方法において、複数の文書構造を持つ文書をまとめて登録する処理には、（１）構造インデクス中の複数の構造に出現しうる各構造の名称と構造の種別の対応を記述した種別定義テーブルを参照し、要素型名からその構造の種別を取得する種別取得ステップと、（２）文書の最上位構造と同じ種別を持つ最上位構造を持つ構造インデクスを取得する最上位構造判定ステップと、（３）構造化文書の登録時に、複数の文書構造を持つ文書の構造インデクスの上位に各構造インデクスをまとめる親ノード（ルートメタノード）を設けることにより、複数の構造インデクスを１つのメタ構造インデクスにまとめるルートメタノード作成ステップを有する。

【００３７】もしくは、（１）構造インデクス中の複数の構造に出現しうる各構造の名称と構造の種別の対応を記述した種別定義テーブルを参照し、要素型名からその構造の種別を取得する種別取得ステップと、（４）登録文書を構造解析して得られる解析済み文書データに、全

文書に共通な仮の最上位構造を追加するステップを有する。

【００３８】種別定義テーブルは、あらかじめ、人手により作成、あるいはシソーラスなどを利用して、同義語を同じ種別に割り当てるなどの方法により自動的に作成する。

【００３９】さらに、本発明による文書検索方法において、構造インデクス中の多くの位置に出現する同じ種別の構造を指定した構造指定検索を効率的に実現するために、文書登録プログラムには、（１）文書登録時に構造インデクスを作成すると共に、別名構造インデクスを生成するステップを有する。

【００４０】別名構造インデクスとは、例えば、作成日・更新日など、各文書構造毎に設定されうる情報を構造インデクスを辿らなくても、一括して検索できるように作成した構造インデクスである。別名構造インデクスで得られる種別を指定して構造指定検索することで、別名構造インデクスから、別名に対応する構造インデクス中の複数の構造を一括して得ることができるため、構造インデクスを辿って指定された構造の文脈識別子を得るよりも、効率的な検索を実現することが可能である。

【００４１】

【発明の実施の形態】（第一の実施例）以下、本発明を適用した第一の実施例について、図面を用いて説明する。

【００４２】はじめに、本実施例のシステム構成について説明する。

【００４３】図１は、本発明による文書検索システムの第一の実施例の全体構成を示す図である。図１に示すとおり、本発明による文書検索システムの第一の実施例は、文書登録サブシステム１０１、文書検索サーバ１０２、文書検索クライアント１０３および１０４、ネットワーク１０５から構成される。

【００４４】文書登録サブシステム１０１は、検索対象として入力される各文書の構造を解析し、検索時に必要となるインデックスデータを作成する。このインデックスデータは、ネットワーク１０５を介して文書検索サーバ１０２に転送され、後に文書検索サーバ１０２が構造検索処理を行う際に用いられる。

【００４５】文書検索サーバ１０２は、検索クライアント１０３および１０４からの検索コマンドを受け取り、文書登録サブシステム１０１が作成したインデックスデータを用いて該検索コマンドの指定する条件に適合する文書内容の検索を行い、検索結果データを要求元の検索クライアントに送り返す。

【００４６】検索クライアント１０３および１０４は、ユーザが対話的に検索条件を指定するための画面をディスプレイ上に表示し、この画面上でユーザが指定した検索条件を、文書検索サーバ１０２によって解釈可能な検索コマンドの形に変換し、この検索コマンドをネットワーク１０５を介して文書検索サーバ１０２に送信する。前記のとおり文書検索サーバ１０２が検索コマンドに対応する検索処理を行い、検索結果データを送り返して来ると、検索クライアント１０２は受け取った検索結果データを画面に表示してユーザに提示する。なお、図１には２台のコンピュータ１０３および１０４を検索クライアントとして使用する構成例を示したが、検索クライアントは１台のみとする構成をとることも、３台以上とする構成をとることもできる。

【００４７】ネットワーク１０５は、ローカルエリアネットワークおよび／または広域ネットワークであって、文書登録サブシステム１０１、文書検索サーバ１０２、検索クライアント１０３および１０４が各種データやコマンドを交換するために用いられる。

【００４８】ここで、図１では文書登録サブシステム１０１から文書検索サーバ１０２にインデックスデータを転送するためにネットワーク１０５を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。あるいは、文書登録サブシステム１０１と文書検索サーバ１０２を１台のコンピュータ上に実装し、データ転送を行わない構成をとることもできる。

【００４９】さらに、図１では検索クライアント１０３および１０４と文書検索サーバ１０２には別個のコンピュータを使用するものとしたが、１個以上の検索クライアントを文書検索サーバと同一のコンピュータ上で実行する構成をとることもできる。

【００５０】以下、本実施例における文書登録サブシステム、すなわち図１の１０１について説明する。

【００５１】図２は、本実施例における文書登録サブシステム１０１の構成を示す図である。

【００５２】図２に示す文書登録サブシステム１０１は、ディスプレイ２０１、キーボード２０２、中央処理装置（ＣＰＵ）２０３、フロッピーディスクドライブ２０４、フロッピーディスク２０５、通信制御装置２０６、主メモリ２０７、磁気ディスク装置２０８、システムバス２０９から構成される。

【００５３】ディスプレイ２０１は、本サブシステムにおける処理の実行状況等を表示するために使用する。キーボード２０２は、文書登録処理の実行等を指示するコマンドを入力するために使用する。中央処理装置２０３は、本サブシステムを構成する各種プログラムを実行する。フロッピーディスクドライブ２０４は、フロッピーディスク２０５に対するデータの読み書きのために使用する。フロッピーディスク２０５は、予め登録対象文書を格納しておき、これを本サブシステムに入力するために使用する。通信制御装置２０６は、ネットワーク１０５を介して前記文書検索サーバ１０２と通信し、リクエストおよびデータの交換を行うために使用する。主メモリ２０７は、本サブシステムによる処理を実行するため

の各種プログラムおよび一時的なデータを保持するために使用する。磁気ディスク装置２０８は、登録された文書データおよび本サブシステムが生成するインデックスデータを格納するために使用する。システムバス２０９は、これらの各種装置を接続するために使用する。

【００５４】主メモリ２０７中には、文書構造解析プログラム２１０、構造インデックス作成プログラム２１１、構造化全文データ生成プログラム２１２、文字列インデックス作成プログラム２１３、文書登録制御プログラム２１４およびシステムプログラム２１５が格納され、またワークエリア２１６が保持される。磁気ディスク装置２０８中には、解析済み文書データ格納領域２１７、構造インデックス格納領域２１８、構造化全文データ格納領域２１９および文字列インデックス格納領域２２０が確保される。

【００５５】文書構造解析プログラム２１０は、ＳＧＭＬを用いて記述され、フロッピーディスク２０５に格納された登録対象文書を読みだし、その論理構造を解析して解析済み文書データを生成し、これを解析済み文書データ格納領域２１７に格納する。構造インデックス作成プログラム２１１は、前記解析済み文書データの持つ論理構造に関する情報を、構造インデックス格納領域２１８に格納されている構造インデックスに登録して構造インデックスを更新する。構造化全文データ生成プログラム２１２は、前記解析済み文書データから前記登録対象文書に関する構造化全文データを生成し、これを構造化全文データ格納領域２１９に格納する。文字列インデックス作成プログラム２１３は、前記構造化全文データから、所定の部分文字列と、該部分文字列の構造化文字位置情報との対応関係を表わすデータを生成し、これを文字列インデックス格納領域２２０に格納されている文字列インデックスに登録して文字列インデックスを更新する。

【００５６】文書登録制御プログラム２１４は、文書構造解析プログラム２１０、構造インデックス作成プログラム２１１、構造化全文データ生成プログラム２１２および文字列インデックス作成プログラム２１３の起動および実行制御を行うとともに、これらのプログラムによって生成された解析済み文書データ、構造インデックスおよび文字列インデックスをネットワーク１０５を介して前記文書検索サーバ１０２に転送する。システムプログラム２１５は、周辺装置との間のデータの入出力など、コンピュータ上で本サブシステムを構成する各プログラムを実行するための基本機能を提供する。ワークエリア２１６は、プログラムの実行時に一時的に必要となるデータを記憶するために用いられる。

【００５７】なお、本実施例ではフロッピーディスク２０５に格納された登録対象文書を入力として読み込む構成としたが、光磁気ディスク、追記型光ディスク等、他種の可搬型媒体から読み込む構成をとることもでき、ネットワーク１０５を介して転送されてくる文書を入力とする構成をとることもできる。さらに、本実施例では生成された解析済み文書データ、構造インデックスおよび文字列インデックスを文書検索サーバ１０２に転送するためにネットワーク１０５を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。あるいは、文書登録サブシステム１０１と文書検索サーバ１０２を１台のコンピュータ上に実装し、データ転送を行わない構成をとることもできる。

【００５８】次に、本実施例における文書登録処理の手順について説明する。

【００５９】図３は、本発明の第一の実施例における文書登録処理の概略手順を示すＰＡＤ（Problem Analysis Diagram）図である。キーボード２０２からの登録指示コマンド等により文書登録制御プログラム２１４が起動されると、本プログラムはまずフロッピーディスク２０５に格納されている登録対象文書の有無とその数を調べ、すべての登録対象文書について、ステップ３０２から３０５までに示す一連の処理を繰り返し実行する（ステップ３０１）。

【００６０】ステップ３０２では、フロッピーディスク２０５に格納されている登録対象文書群の中から、未処理の登録対象文書を１個選択して読み込む。ステップ３０３では、読み込んだ登録対象文書に文書識別子を割り当てる。文書識別子は、文書データベース中で特定の文書を一意に識別する番号である。

【００６１】ステップ３０４では、この登録対象文書を入力として文書構造解析プログラム２１０を実行する。ここで、文書構造解析プログラム２１０は、この登録対象文書に対応する解析済み文書データを生成して解析済み文書データ格納領域２１７に格納する。

【００６２】ステップ３０５では、ステップ３０４で生成した解析済み文書データを入力として構造インデックス作成プログラム２１１を実行する。構造インデックス作成プログラム２１１は、まず構造インデックス格納領域２１７から現時点での構造インデックスを読み出し、与えられた解析済み文書データの持つ構造情報をこの構造インデックスに登録し、更新された構造インデックスを再び構造インデックス格納領域２１８に格納する。

【００６３】ステップ３０６では、ステップ３０４で生成した解析済み文書データを入力として、構造化全文データ生成プログラム２１２を実行する。構造化全文データ生成プログラム２１２は、与えられた解析済み文書データを参照して、ステップ３０３で読み込んだ登録対象文書に対応する構造化全文データを生成し、構造化全文データ格納領域２１９に格納する。

【００６４】ステップ３０７では、ステップ３０６で生成した構造化全文データを入力として文字列インデックス作成プログラム２１３を実行する。文字列インデック

ス作成プログラム２１３は、まず文字列インデックス格納領域２２０から現時点での文字列インデックスを読み出し、構造化全文データから、所定の部分文字列と、該部分文字列の構造化文字位置情報との対応関係を表わすデータを生成してこれを文字列インデックスに登録し、更新された文字列インデックスを再び文字列インデックス格納領域２２０に格納する。

【００６５】すべての登録対象文書についてステップ３０２から３０７までに示す一連の処理が終了すると、文書登録制御プログラム２１４はステップ３０８を実行して処理を終了する。ステップ３０８では、解析済み文書データ格納領域２１７に格納されたすべての解析済み文書データ、構造インデックス格納領域２１８に格納された構造インデックス、および文字列インデックス格納領域２２０に格納された文字列インデックスを、ネットワーク１０５を介して文書検索サーバ１０２に転送する。

【００６６】次に、図３におけるステップ３０４の詳細、すなわち本実施例における文書構造解析プログラム２１０の処理手順について説明する。

【００６７】ここで、文書構造解析プログラム２１０は、ＳＧＭＬを用いて記述された１個の登録対象文書を対象として構造解析処理を行う。ＳＧＭＬでは、特定の種別に属する文書群に共通する論理構造を、ＤＴＤ（Document Type Definition）によって定義する。図４がＤＴＤの一例である。ＤＴＤは、文書を構成する論理要素（以下、これを単に「要素」と呼ぶ）の集合を定義することによって、文書の論理的構造を定義する。図４において、文字列“<!ELEMENT”と文字列“>”で囲まれた部分を要素型宣言と呼び、１個の要素型宣言が、１種類の要素型に属する要素群が共通して持つ名前（これを要素型名と呼ぶ）とその構造を規定する。要素型宣言中の左側に示されている文字列が要素型名、右側に示されている部分がその内容がとる構造の定義である。

【００６８】図４に示すＤＴＤにおいて、要素型“論文”に関する要素型宣言は、この要素型に属する要素の内容が、“タイトル”、“執筆者”、“日付”、“本文”および“文献リスト”という要素型に属する要素１個ずつを、この順序に従って並べた構造を持つことを規定している。ここで、複数の要素型名を文字“,”で区切って並べることにより、それらの要素型名に属する要素が指定した順序で出現しなければならないことを表現する。

【００６９】要素型“執筆者”に関する要素型宣言は、この要素型に属する要素の内容が、要素型“名前”に属する要素の１個以上の繰り返しからなる構造を持つことを規定している。ここで、要素型名の後ろに文字“+”を付加することにより、その要素型名に属する要素が１個以上出現することを表現する。

【００７０】要素型“本文”に関する要素型宣言は、この要素型に属する要素の内容が、要素型“章”に属する

要素の０個以上の繰り返しからなる構造を持つことを規定している。ここで、要素型名の後ろに文字“*”を付加することにより、その要素型名に属する要素が０個以上出現することを表現する。

【００７１】要素型“章”に関する要素型宣言は、この要素型に属する要素の内容が、要素型“章題”に属する要素１個の後ろに、要素型“段落”または“備考”に属する要素を０個以上続け、さらにその後ろに要素型“節”に属する要素を０個以上繰り返した構造を持つことを規定している。ここで、複数の要素型名を文字“|”で区切って並べることにより、それらのいずれかの要素型に属する要素が出現することを表現する。

【００７２】要素型“節”に関する要素型宣言は、この要素型に属する要素の内容が、要素型“節題”に属する要素１個の後ろに、要素型“段落”または“備考”に属する要素を０個以上続け、さらにその後ろに要素型“項”に属する要素を０個以上繰り返した構造を持つことを規定している。

【００７３】要素型“項”に関する要素型宣言は、この要素型に属する要素の内容が、要素型“項題”に属する要素１個の後ろに、要素型“段落”または“備考”に属する要素を０個以上繰り返した構造を持つことを規定している。

【００７４】要素型“文献リスト”に関する要素型宣言は、この要素型に属する要素の内容が、要素型“文献”に属する要素の１個以上の繰り返しからなる構造を持つことを規定している。

【００７５】要素型“文献”に関する要素型宣言は、この要素型に属する要素の内容が、“タイトル”、“執筆者”、“日付”および“出典”という要素型に属する要素１個ずつを、この順序に並べた構造を持つことを規定している。

【００７６】また、要素型“タイトル”、“名前”、“日付”、“章題”、“節題”、“項題”、“強調”および“出典”に属する要素の内容は、単に“#PCDATA”と規定されている。これは、これらの要素がそれ以上の下位構造をもたず、単なる文字の列からなる内容を持つことを規定している。要素型“段落”および“備考”に関する要素型宣言は、これらの要素型に属する要素が、要素型“強調”に属する要素または単なる文字列を０個以上繰り返した構造を持つことを規定している。

【００７７】ＤＴＤ中において、文字列“<!ATTLIST”と文字列“>”で囲まれた部分を属性リスト宣言と呼び、１個の属性リスト宣言が、１種類の要素型に属する要素群が共通して持つ属性を定義する。図４に示すＤＴＤでは、要素型“備考”に属する要素が、属性“type”を持つこと、この属性は“参照”または“注釈”のいずれかを値としてとることができ、指定が省略された場合には“参考”が値として与えられることを規定している。

19

【００７８】図４に示すＤＴＤに従って記述されたＳＧＭＬ文書の一例を、図５に示す。文書先頭の、文字列“<!DOCTYPE”と文字列“>”で囲まれた部分を文書型宣言と呼び、そのＳＧＭＬ文書が従うＤＴＤと、最上位要素の要素型名を宣言する。図５に示した例では、この文書がファイル“ronbun.dtd”に格納されているＤＴＤに従い、最上位要素の要素型名は“論文”であることが規定されている。ここでは、ファイル“ronbun.dtd”に図４に示した前記ＤＴＤが格納されているものとする。

【００７９】図５に示すとおり、ＳＧＭＬでは文書を構成する個々の要素について、その先頭位置と末尾位置を示すマークを付加することにより、文書構造を明示的に記述する。各要素の先頭位置を示すマークを“開始タグ”、終了位置を示すマークを“終了タグ”と呼ぶ。開始タグは、文字列“<”と“>”との間に、その要素の要素型名を記述することによって示す。終了タグは、文字列“</”と“>”との間に、その要素の要素型名を記述することによって示す。要素が属性を持つ場合、属性値の指定を開始タグ中（要素型名の後ろ）に記述することができる。属性値の指定は、属性名と属性値との間に文字列“=”を置くことによって示す。例えば図５において、開始タグ“<備考 type=注釈>”は、要素“備考”の属性“type”に属性値“注釈”を与えている。ＳＧＭＬ文書中において、これらのタグを用いて文書構造を記述している部分を「文書インスタンス」と呼ぶ。

【００８０】図３におけるステップ３０４の詳細、すなわち本実施例における文書構造解析プログラム２１０の処理手順を示すＰＡＤ図を、図７に示す。

【００８１】図７に示すとおり、文書構造解析プログラム２１０は、ＳＧＭＬを用いて記述された１個の登録対象文書を入力として起動されると、まず該文書の先頭に記述された文書型宣言を読み、その構文を解析する（ステップ７０１）。次に、ステップ７０２において、文書型宣言中における構文エラーの有無を判定する。構文エラーが検出された場合、ステップ７０３に進み、エラーメッセージを出力して処理を中断する。

【００８２】文書型宣言に構文エラーがない場合にはステップ７０４に進み、その文書型宣言で指定されているＤＴＤファイルが存在するかどうかを判定する。ＤＴＤファイルが検出されない場合、ステップ７０５に進み、エラーメッセージを出力して処理を中断する。

【００８３】ＤＴＤファイルが検出された場合にはステップ７０６に進み、該ファイルの内容を読み込んでその構文解析を行う。次に、ステップ７０７において、ＤＴＤ中における構文エラーの有無を判定する。構文エラーが検出された場合、ステップ７０８に進み、エラーメッセージを出力して処理を中断する。構文エラーが検出されなかった場合にはステップ７０９に進み、ＤＴＤが定義する文書構造モデルを記述したデータである文書構造テーブルをメモリ上に生成する。

20

【００８４】次に、ステップ７１０において、前記文書構造テーブルを参照しながら文書インスタンスを読み込み、構造解析を行い、その結果として解析済み文書データを生成する。次に、ステップ７１１において、文書インスタンス中に構文エラーや構造エラー（ＤＴＤが定義する構造モデルからの逸脱）がないかどうかを判定する。構文エラーまたは構造エラーが検出された場合にはステップ７１２に進み、エラーメッセージを出力して処理を中断する。エラーが検出されなかった場合にはステップ７１３に進み、前記登録対象文書を識別する文書識別子と、前記ステップ７１０における構造解析によって得られた解析結果データとからなる解析済み文書データを、解析済み文書データ格納領域２１７に出力して処理を終了する。

【００８５】ここで一例として、図５に示すＳＧＭＬ文書を登録対象文書として文書構造解析プログラム２１０を実行し、該文書が参照するＤＴＤファイル“ronbun.dtd”の内容が図４に示すＤＴＤだった場合について説明する。この場合、ステップ７０９において生成される文書構造テーブルは、図８に示すようなデータ構造をとる。図８に示すとおり、文書構造テーブルは構造定義と属性定義の二つの部分からなる。構造定義は、ＤＴＤを構成する各要素型の要素型名に対応させて、その要素型に属する要素がとりうる内容のデータモデルを定義する。また属性定義は、ＤＴＤを構成する各要素型の要素型名に対応させて、その要素型に属する要素が持つ各属性の属性名、属性値のタイプおよびデフォルト値を定義する。この構造定義を参照することにより、文書インスタンス中に現れる要素群の並び順や階層関係が正しいかどうか（構造エラーの有無）を判断し、また省略されたタグや属性値指定があった場合、それらを補足することができる。

【００８６】図５に示すＳＧＭＬ文書が登録対象文書として文書構造解析プログラム２１０への入力として与えられ、そのＤＴＤが図４に示すものだった場合、図６に示す木構造データが解析済み文書データとして得られる。図６は、図５に示すＳＧＭＬ記述によって表現されている文書の論理構造を、図形的に示した模式図である。図６に示すように、構造化文書の持つ論理的構造は、個々の要素を中間節、文字列データを終端節とする木構造としてとらえることができる。図６では、各要素を楕円形、文字列データを矩形で表わしている。

【００８７】なお、本実施例ではＳＧＭＬを用いて記述された構造化文書を登録対象文書として処理する構成をとったが、ＯＤＡ（Open Document Architecture）など、他の形式によって記述された構造化文書を登録対象文書とするように構成することもできる。

【００８８】図９は、図３におけるステップ３０５の詳細、すなわち本実施例における構造インデックス作成プログラム２１１の処理手順を示すＰＡＤ図である。

【００８９】構造インデックス作成プログラム２１１は、まずステップ９０１において、既存の構造インデックスが構造インデックス格納領域２１８中に存在しているかどうかを判定する。該領域中に構造インデックスが存在しない場合にはステップ９０２に進み、初期状態（空）の構造インデックスを生成する。既存の構造インデックスを検出した場合にはステップ９０３に進み、該構造インデックスを読み込む。

【００９０】次に、ステップ９０４において、解析済み文書データ格納領域２１７から登録対象文書の解析済み文書データを読み込む。

【００９１】次に、ステップ９０５において、前記解析済み文書データの木構造を構成するすべてのノード（要素および文字列データ）を対象として、ステップ９０６からステップ９０９に示す処理を繰り返す。

【００９２】ステップ９０６では、解析済み文書データ中で現在着目しているノードについて、構造インデックス中に、該ノードに対応するメタノード（メタ要素またはメタ文字列データ）が存在するかどうかを判定する。対応メタノードが存在しない場合はステップ９０７に進み、該ノードに対応するメタノードを生成して構造インデックスに登録し、さらに該登録したメタノードに対して、該メタノードを構造インデックス中で一意に識別する番号である文脈識別子を割り当てる（ステップ９０８）。ステップ９０９では、解析済み文書データ中で現在着目しているノードと、構造インデックス中で該ノードに対応するメタノードを識別する文脈識別子との対応関係を、解析済み文書データに付加して該解析済み文書データを更新する。

【００９３】ステップ９０５以下の繰り返しを終了するとステップ９１０に進み、前記更新された解析済み文書データを出力して解析済み文書データ格納領域２１７に格納する。次に、ステップ９１１では、前記更新された構造インデックスを出力して構造インデックス格納領域２１８に格納し、処理を終了する。

【００９４】ここで、前記ステップ９０５において解析済み文書データの木構造を構成するすべてのノードを対象とする繰り返し処理を行う際に、該木構造を辿り、個々のノードを処理していく順序を図１０を用いて説明する。図１０では、楕円形で要素ノード、矩形で文字列ノードを表現し、あるノードが複数の下位ノードを持つ場合には、各下位ノードをその出現順に左から右に並べて表現している。また、各ノードに表示された数字がその処理順序を示している。図１０に示すとおり、ステップ９０５においてノード群を処理していく順序は、木構造の根に位置するノードから出発し、ある特定のノードとその下位ノード群を処理する際には、まず該ノードの処理を行い、次に該ノードの下位ノード群を、その出現順に従って処理していく順序となる。

【００９５】次に、前記ステップ９０６における処理、

すなわち解析済み文書データ中で現在着目しているノードについて、構造インデックス中に、該ノードに対応するメタノードが存在するかどうかを判定する処理の内容を、図１１を用いて説明する。図１１は、図の左側に示す解析済み文書データの木構造を構成するノード群と、右側に示す構造インデックスの木構造を構成するノード（メタノード）群との対応関係を示す図である。

【００９６】ここで、本実施例では、解析済み文書の木構造中におけるあるノードの木構造アドレスと、構造インデックスの木構造中におけるあるメタノードの木構造アドレスとが等しい場合に、該ノードと該メタノードとは対応するものと定義する。ここでいう木構造アドレスとは、木構造の根から出発して上位ノードから下位ノードへと辿り、特定のノードに至るまでの道筋を、該道筋上に存在する各ノードの種別（要素か文字列データか、要素の場合、どの要素型に属するか）と、それらのノードが種別の等しい兄弟ノード群中で何番目に現れたかを示す番号との組み合わせによって表現するアドレスのことである。

【００９７】例えば、図１１に示す解析済み文書データ中のノード群のうち、ノード１１０１は上位ノードをもたず、兄弟ノード中では最初の“論文”要素ノードであるから、その木構造アドレスは“/論文[1]”と表記することができる。同様に、ノード１１０２はノード１１０１の下位ノードであり、兄弟ノード中では最初の“章”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]”と表記することができる。また、ノード１１０３はノード１１０２の下位ノードであり、兄弟ノード中では２番目の“節”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]/節[2]”と表記することができる。また、ノード１１０４はノード１１０３の下位ノードであり、兄弟ノード中では最初の“段落”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]/節[2]/段落[1]”と表記することができる。

【００９８】同様にして、図１１右側の構造インデックスの木構造を構成する各メタノードについてその木構造アドレスを求めてみると、メタノード１１０５の木構造アドレスは“/論文[1]”となり、ノード１１０１の木構造アドレスと等しくなる。同様に、メタノード１１０６の木構造アドレスは“/論文[1]/章[1]”となってノード１１０２の木構造アドレスと等しく、メタノード１１０７の木構造アドレスは“/論文[1]/章[1]/節[2]”となってノード１１０３の木構造アドレスと等しくなる。よって、前記ステップ９０６において、ノード１１０１はメタノード１１０５と、ノード１１０２はメタノード１１０６と、ノード１１０３はメタノード１１０７とそれぞれ対応するものと判定される。

【００９９】なお、図１１の構造インデックス中にはノード１１０４と等しい木構造アドレスを持つメタノードはないので、ノード１１０４と対応するメタノードは構

造インデックス中に存在しないものと判定され、前記ステップ９０７において新たなメタノードが生成され、該メタノードが構造インデックスに登録されることになる。前記ステップ９０７において、あるノードに対応する新たなメタノードを登録する際には、該ノードの上位ノードに対応するメタノードの持つ下位ノード群の末尾に、該ノードに対応する種別のメタノードを追加する。図１１のノード１１０４に対応するメタノードを登録する場合には、ノード１１０４の上位ノードであるノード１１０３に対応するメタノード１１０７の下位に、要素型“段落”のメタノードが追加され、該メタノードは兄弟メタノード群の末尾に置かれることになる。

【０１００】次に、複数の解析済み文書データを順次重ね合わせることにより構造インデックスを生成していく過程について、図１２を用いて説明する。図１２において、１２０１、１２０３および１２０５は、それぞれ登録対象文書の解析済み文書データを表わしている。これらの解析済み文書データの構造を既存の構造インデックス上に順次重ね合わせることにより、構造インデックスが形成されていく。まず最初に文書１の解析済み文書データ１２０１が入力されると、最初の段階では構造インデックスは初期状態（空）であるため、該解析済みデータと等価な木構造が生成されてそのまま構造インデックスに登録され、構造インデックスは１２０２に示す状態となる。新たに生成されたメタ要素にはＥ１からＥ５までの文脈識別子、新たに生成されたメタ文字列データにはＣ１からＣ３までの文脈識別子が割り当てられる。

【０１０１】次に文書２の解析済み文書データ１２０３が入力されると、既存の構造インデックス（１２０２）と構造が重複する部分については何も行わず、１２０２上に対応する部分がなかった部分構造（図中の斜線部分）だけが新たに登録される。新たに生成されたメタ要素には文脈識別子Ｅ６およびＥ７、新たに生成されたメタ文字列データには文脈識別子Ｃ４が割り当てられる。次に文書３の解析済み文書データ１２０５が入力されると、既存の構造インデックス（１２０４）と構造が重複する部分については何も行わず、１２０４上に対応する部分がなかった部分構造（図中の斜線部分）だけが新たに登録される。新たに生成されたメタ要素には文脈識別子Ｅ８、Ｅ９およびＥ１０、新たに生成されたメタ文字列データには文脈識別子Ｃ５およびＣ６が割り当てられる。このようにして、３個の文書が登録された段階で、構造インデックスは１２０６に示す状態となる。

【０１０２】図１３は、図３におけるステップ３０６の詳細、すなわち本実施例における構造化全文データ生成プログラム２１２の処理手順を示すＰＡＤ図である。

【０１０３】構造化全文データ生成プログラム２１２は、まずステップ１３０１において、解析済み文書データ格納領域２１７から前記登録対象文書の解析済み文書データを読み込む。

【０１０４】次に、ステップ１３０２において、前記登録対象文書を識別する文書識別子を構造化全文データ格納領域２１９に出力する。

【０１０５】次に、ステップ１３０３において、前記解析済み文書データの木構造を構成するすべてのノード（要素ノードおよび文字列データノード）を対象として、ステップ１３０４からステップ１３０６に示す処理を繰り返す。

【０１０６】ステップ１３０４では、解析済み文書データ中で現在着目しているノードについて、該ノードが要素ノードであるか文字列データノードであるかを判定し、該ノードが文字列データノードである場合に限りステップ１３０５に進む。ステップ１３０５では、前記解析済み文書データから、前記現在着目している文字列データノードに対応する文脈識別子を求め、該文脈識別子を構造化全文データ格納領域２１９に出力する。次に、ステップ１３０６で、前記現在着目している文字列データノードの内容文字列を構造化全文データ格納領域２１９に出力する。

【０１０７】ステップ１３０３以下の繰り返しをすべて終了した段階で本プログラムは処理を終了する。

【０１０８】図１４に、構造化全文データ生成プログラム２１２が出力する構造化全文データのファイル形式を示す。図１４は、図５に示したＳＧＭＬ文書を入力として構造化全文データを生成した場合について例示している。図１４に示すとおり、本実施例における構造化全文データのデータファイルは、先頭に文書識別子を記述し、その後ろに、文脈識別子と該文脈識別子に対応する内容文字列との対を、文書中に存在する文字列データの数だけ繰り返した構造をとる。

【０１０９】例えば、図１４に示す構造化全文データに対応する登録対象文書の文書識別子は“D1”であり、図５において“日付”要素の内容として記述されていた文字列データには文脈識別子“C5”が与えられている。なお、図１４他ではこれらの識別子を記号により表現しているが、文書識別子としてデータ中に実際に記録される値は登録対象文書の集合中で特定の文書を一意に識別する番号（整数値）であり、文脈識別子の値は構造インデックスを構成するメタノードの集合中で特定のメタノードを一意に識別する番号（整数値）である。

【０１１０】図１５は、図３におけるステップ３０７の詳細、すなわち本実施例における文字列インデックス作成プログラム２１３の処理手順を示すＰＡＤ図である。

【０１１１】文字列インデックス作成プログラム２１３は、まずステップ１５０１において、既存の文字列インデックスが文字列インデックス格納領域２２０中に存在しているかどうかを判定する。該領域中に文字列インデックスが存在しない場合にはステップ１５０２に進み、初期状態（空）の文字列インデックスを生成する。既存の文字列インデックスを検出した場合にはステップ１５

０３に進み、該文字列インデックスを読み込む。

【０１１２】次に、ステップ１５０４において、登録対象文書の構造化全文データを構造化全文データ格納領域２１９から読み込む。

【０１１３】次に、ステップ１５０５において、前記構造化全文データを構成するすべての内容文字列を対象として、ステップ１５０６から１５０７に示す処理を繰り返す。

【０１１４】ステップ１５０６では、構造化全文データ中で現在着目している内容文字列から、所定の部分文字列を抽出する。ステップ１５０７では、前記ステップ１５０６で抽出した各部分文字列と該部分文字列の構造化文字位置情報との対応関係を、前記文字列インデックスに登録する。

【０１１５】ステップ１５０５以下の繰り返しを終了するとステップ１５０８に進み、不要となった構造化全文データを構造化全文データ格納領域２１９から削除して破棄する。次に、ステップ１５０９において、更新された文字列インデックスを出力して文字列インデックス格納領域２２０に格納し、処理を終了する。

【０１１６】ここで、前記ステップ１５０６において、ある内容文字列から所定の部分文字列を抽出する際には、予め抽出すべき部分文字列の長さを定めておき、対象とする内容文字列の先頭から出発して、開始位置を１ずつ増しながら前記予め定めておいた長さの部分文字列を順次抽出する。例えば、抽出する部分文字列の長さを２文字とし、処理対象として図１４に示す内容文字列群のうち文脈識別子Ｃ１２９に対応する内容文字列“変換処理の実例”を用いた場合、抽出される部分文字列は“変換”、“換処”、“処理”、“理の”、“の実”および“実例”の６個となる。

【０１１７】更に、内容文字列の末尾部分については、長さ１文字から部分文字列長－１文字までの各部分文字列をも抽出する。前記の例では“例”が抽出される。これらの部分文字列を前記ステップ１５０７において文字列インデックスに登録する際には、各部分文字列と、それらが出現した位置を表わす構造化文字位置情報との対応関係を登録する。ここで、構造化文字位置情報は、対応する部分文字列を含む文書の文書識別子、該文書中において前記部分文字列を含む文字列データの文書構造中における位置を識別する文脈識別子、および文書中における前記部分文字列の先頭文字位置から構成される。

【０１１８】図１６に、本実施例における文字列インデックスのデータ構造を示す。図１６は、文字列インデックス作成プログラム２１３を用いて図１４に示す構造化全文データを処理し、文字列インデックスに該構造化全文データに含まれる部分文字列群の登録を終えた段階において、該文字列インデックスのデータ構造の一部（前記内容文字列“変換処理の一例”に関連する部分）を図示したものである。ただし、図１６では、前記内容文字列の末尾“例”に対応する文字ノードおよび構造化文字位置情報は省略している。また、前記内容文字列の直前に位置する文字の文字位置を“Ｘ”として、相対的に文字位置を表現している。

【０１１９】図１６に示すとおり、文字列インデックスは、登録対象文書中に現れる所定長さの部分文字列すべてについて、その出現位置情報（文書識別子、文脈識別子および先頭文字位置の組み合わせからなる構造化文字位置情報）のリストを保持する。ここで、インデックス検索を高速化するため、１文字目が共通する部分文字列群については、１文字目の情報をすべての部分文字列が共有するデータ構造をとり、また文字列インデックスのルートから１文字目のノードへのポインタの並びは、ポインタが指す文字の文字コード順となるように配列する。同様に、１文字目のノードから２文字目のノードへのポインタの並びも、ポインタが指す文字の文字コード順となるように配列する。

【０１２０】文書データベースに登録するすべての登録対象文書を処理してその内部に現れる部分文字列群を文字列インデックスに登録しておけば、この文字列インデックスを参照するだけで（文書データ本体を一切走査することなく）任意の２文字からなる文字列がどの文書中のどの位置に出現するかを知ることができる。（2文字以外の長さの文字列を検索する方法については後述する。）なお、本実施例では、予め定める部分文字列の長さを２文字としたが、該長さをこれ以外の値としても同様な文字列インデックスを構築することができる。また、本実施例では部分文字列の長さを固定長としたが、この長さを可変として同様な文字列インデックスを構築することもできる。

【０１２１】以上が、本実施例における文書登録サブシステム１０１の説明である。

【０１２２】以下、本発明の第一の実施例における文書検索サーバ、すなわち図１の１０２について説明する。

【０１２３】図１７は、本実施例における文書検索サーバ１０２の構成を示す図である。

【０１２４】図１７に示す文書検索サーバ１０２は、ディスプレイ２０１、キーボード２０２、中央処理装置（ＣＰＵ）２０３、通信制御装置２０６、主メモリ２０７、磁気ディスク装置２０８、システムバス２０９から構成される。

【０１２５】ディスプレイ２０１は、本サーバの稼動状況等を表示するために使用する。キーボード２０２は、本サーバの起動・停止等を指示するコマンドを入力するために使用する。中央処理装置２０３は、本サーバを構成する各種プログラムを実行する。通信制御装置２０６は、ネットワーク１０５を介して前記文書登録サブシステム１０１および前記検索クライアント（１０３および１０４）と通信し、リクエストおよびデータの交換を行うために使用する。主メモリ２０７は、本サーバによる

処理を実行するための各種プログラムおよび一時的なデータを保持するために使用する。磁気ディスク装置２０８は、文書データベースを構成する文書データ群および本サーバが文書検索時に参照するインデックスデータを格納するために使用する。システムバス２０９は、これらの各種装置を接続するために使用する。

【０１２６】主メモリ２０７中には、検索条件解析プログラム１７０１、文字列インデックス検索プログラム１７０２、文書検索制御プログラム１７０３およびシステムプログラム２１５が格納され、またワークエリア２１６が保持される。磁気ディスク装置２０８中には、解析済み文書データ格納領域２１７、構造インデックス格納領域２１８、文字列インデックス格納領域２２０および検索結果データ格納領域１７０４が確保される。

【０１２７】検索条件解析プログラム１７０１は、検索クライアント１０３および１０４から受信した検索リクエスト中に含まれる検索条件式を解析し、文字列インデックス検索プログラム１７０２によって直接検索可能な条件指定に翻訳する。文字列インデックス検索プログラム１７０２は、検索条件解析プログラム１７０１によって翻訳された条件指定に従って、文字列インデックス格納領域２２０に格納されている文字列インデックスを検索し、得られた検索結果データを検索結果データ格納領域１７０４に格納する。

【０１２８】文書検索制御プログラム１７０３は、検索条件解析プログラム１７０１および文字列インデックス検索プログラム１７０２の起動および実行制御を行うとともに、ネットワーク１０５を介して、文書登録サブシステム１０１および検索クライアント（１０３および１０４）との間でリクエストおよびデータの交換を行う。システムプログラム２１５は、周辺装置との間のデータの入出力など、コンピュータ上で本サーバを構成する各プログラムを実行するための基本機能を提供する。ワークエリア２１６は、プログラムの実行時に一時的に必要となるデータを記憶するために用いられる。

【０１２９】なお、本実施例では文書登録サブシステム１０１および検索クライアント１０３および１０４との間でデータを転送するためにネットワーク１０５を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。また、文書登録サブシステム１０１と文書検索サーバ１０２を１台のコンピュータ上に実装し、これらの間でデータ転送を行わない構成をとることもできる。また、１個以上の検索クライアントを文書検索サーバ１０２と同一のコンピュータ上で実行し、これらの間でデータ転送を行わない構成をとることもできる。

【０１３０】図１８は、本発明の第一の実施例における文書検索処理の概略手順を示すＰＡＤ図である。キーボード２０２からのサーバ起動コマンド等により文書検索制御プログラム１７０３が起動されると、本プログラムはサーバとして文書登録サブシステム１０１および検索クライアント（１０３、１０４等）からリクエストを受信してはその処理を行うループに入る（ステップ１８０１）。このループは、キーボード２０２からサーバの停止を指示するコマンドが入力されるまで継続する。

【０１３１】ステップ１８０１のループは、文書登録サブシステム１０１および検索クライアント（１０３および１０４）からリクエストを受信する処理（ステップ１８０２）と、受信したリクエストの種別を判定し、該種別に対応する処理に分岐する処理（ステップ１８０３）を繰り返す。

【０１３２】ステップ１８０３では、受信したリクエストの種別を判定し、該リクエストが、文書登録サブシステム１０１から送信されたデータベース更新リクエスト（新たな文書群を登録して文書データベースを更新することを求めるリクエスト）であった場合、ステップ１８０４およびステップ１８０５からなる処理に分岐する。

【０１３３】また、前記リクエストが、検索クライアント（１０３、１０４等）から送信された文書検索リクエスト（特定の検索条件を満たす文書群の検索を求めるリクエスト）であった場合、ステップ１８０６、１８０７および１８０８からなる処理に分岐する。また、前記リクエストが、検索クライアント（１０３、１０４等）から送信された、検索結果問い合わせリクエスト（特定の検索処理の結果を問い合わせるリクエスト）であった場合、ステップ１８０９からなる処理に分岐する。また、前記リクエストが、検索クライアント（１０３、１０４等）から送信された、文書転送リクエスト（指定された文書データの転送を求めるリクエスト）であった場合、ステップ１８１０からなる処理に分岐する。分岐先の処理が終了した後は再びステップ１８０２に戻ってループを継続する。

【０１３４】ステップ１８０４では、文書登録サブシステム１０１から、新規に登録された文書群の解析済み文書データを受信し、該解析済み文書データを解析済み文書データ格納領域２１６に追加する。次に、ステップ１８０５では、文書登録サブシステム１０１から、新規に登録された前記文書群の内容を反映して更新された構造インデックスおよび文字列インデックスを受信し、これらをそれぞれ構造インデックス格納領域２１８および文字列インデックス格納領域２２０に格納する。

【０１３５】ステップ１８０６では、検索条件解析プログラム１７０１を実行し、文書検索リクエスト中で指定されている検索条件を解析し、該検索条件を、文字列インデックス検索プログラム１７０２によって直接処理可能な条件指定（以下、これを展開済み検索条件データと呼ぶ）に変換する。次に、ステップ１８０７では、前記ステップ１８０６によって生成された展開済み検索条件データを入力として文字列インデックス検索プログラム

１７０２を実行し、該展開済み検索条件データが指定する条件を満たす文書群を検索して、検索結果データを求め、該検索結果データを一意に識別する検索結果識別子と対応付けて検索結果データ格納領域１７０４に格納する。次に、ステップ１８０８では、前記検索結果識別子を、要求元の検索クライアントに返送する。

【０１３６】ステップ１８０９では、問い合わせの内容に応じて前記ステップ１８０７で求めた検索結果データの一部または全体を検索結果データ格納領域１７０４から抽出し、要求元の検索クライアントに転送する。

【０１３７】ステップ１８１０では、文書転送リクエスト中で指定されている文書（複数の文書が指定されている場合には指定されている文書すべて）の解析済み文書データを解析済み文書データ格納領域２１７から抽出し、要求元の検索クライアントに転送する。

【０１３８】図１９は、図１８におけるステップ１８０６の詳細、すなわち本実施例における検索条件解析プログラム１７０１の処理手順を示すＰＡＤ図である。

【０１３９】検索条件解析プログラム１７０１は、文書検索リクエスト中で指定されている検索条件を入力として起動されると、まずステップ１９０１において、該検索条件中に構造条件が含まれているかどうかを判定する。そして、構造条件が含まれている場合に限り、ステップ１９０２およびステップ１９０３からなる処理を実行する。構造条件が含まれていなかった場合にはステップ１９０４に進む。

【０１４０】ステップ１９０２では、構造インデックス格納領域２１８から構造インデックスを読み込む。次に、ステップ１９０３では、該構造インデックスを参照して、前記構造条件を満たす構造内に含まれるすべての文字列データの文脈識別子の集合を求める。以下、該集合を文脈識別子集合と呼ぶ。

【０１４１】ステップ１９０４では、前記検索条件中に文字列条件として指定された文字列の長さが、前記文字列インデックスを作成する際に予め定めた部分文字列長を超えているかどうかを判定する。前記指定文字列の長さが前記部分文字列長を超えている場合にはステップ１９０５に進み、前記指定文字列の先頭から、開始文字位置を１ずつ増しつつ、前記部分文字列長に等しい長さの部分文字列群を抽出し、これらの部分文字列を要素とする部分文字列リストを生成する。前記指定文字列の長さが前記部分文字列長を超えていない場合にはステップ１９０６に進み、空の（要素をもたない）部分文字列リストを生成する。

【０１４２】ステップ１９０７では、前記ステップ１９０３で求めた文脈識別子集合、前記検索条件中に含まれていた指定文字列、および前記ステップ１９０５またはステップ１９０６で生成した部分文字列リストから構成される展開済み部分文字列データを生成して処理を終了する。

【０１４３】ここで、図２０は、検索条件解析プログラム１７０１の処理過程における、展開済み解析条件データの生成例を示す図である。

【０１４４】図２０において、２００１は、文書検索リクエスト中で指定された検索条件の一例である。検索条件２００１は、構造条件指定“章/段落[1]”と、文字列条件指定“ガード”とから構成されている。前記検索条件は、“章”要素の直接の下位にある最初の“段落”要素内に、文字列“ガード”が出現するケースを検索すべきことを指定している。

【０１４５】ここで、構造インデックスの内容が２００２に示すとおりであったとすると、前記ステップ１９０３においてこの構造インデックスを参照することにより、前記構造条件指定を満たす“段落”要素の文脈識別子はＥ５およびＥ１４であることがわかる。従って、これらの段落の下位にある文字列データ、すなわち文脈識別子がＣ３またはＣ９である文字列データ内に、文字列“ガード”が出現するケースを検索すればよいことがわかる。ただし、検索に用いる文字列インデックスには、長さ２の部分文字列についてのみその出現位置が登録されているので、３文字からなる前記指定文字列を直接検索することはできない。そこで、ステップ１９０５において、前記指定文字列を分解して長さ２の部分文字列からなるリストを生成する。前記のとおり指定文字列が“ガード”だった場合、抽出される部分文字列は“ガー”および“ード”となる。

【０１４６】この結果、前記ステップ１９０７において、２００３に示す展開済み検索条件データ、すなわち文脈識別子集合が｛Ｃ３，Ｃ９｝、指定文字列が“ガード”、部分文字列リストが｜“ガー”，“ード”｜であるデータが生成される。

【０１４７】図２１は、図１８におけるステップ１８０７の詳細、すなわち本実施例における文字列インデックス検索プログラム１７０２の処理手順を示すＰＡＤ図である。

【０１４８】文字列インデックス検索プログラム１７０２は、前記検索条件解析プログラム１７０１が生成した展開済み検索条件データを入力として起動される。本プログラムは、起動されるとまずステップ２１０１において、文字列インデックス格納領域２２０から文字列インデックスを読み込む。次に、ステップ２１０２に進み、検索結果データを初期化する。

【０１４９】次に、ステップ２１０３では、前記展開済み検索条件データ中に含まれている指定文字列の長さと、前記文字列インデックスを作成する際に予め定めた部分文字列の長さとを比較する。前記指定文字列の長さが前記部分文字列の長さに等しい場合にはステップ２１０４に分岐する。前記指定文字列の長さが前記部分文字列の長さに満たない場合にはステップ２１０５に分岐する。前記指定文字列の長さが前記部分文字列の長さを超

える場合にはステップ２１０６に分岐する。

【０１５０】ステップ２１０４では、文字列インデックス中で前記指定文字列を検索し、該文字列に対応している構造化文字位置の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【０１５１】ステップ２１０５では、文字列インデックス中で前記指定文字列を検索し、該文字列の末尾に対応する文字ノードより前方に存在するすべての構造化文字位置情報の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【０１５２】ステップ２１０６では、前記展開済み検索条件データ中の部分文字列リストを構成する各部分文字列について、ステップ２１０７を繰り返す。ステップ２１０７では、文字列インデックス中で前記部分文字列を検索し、該文字列に対応している構造化文字位置情報の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群を前記部分文字列に対応付けて記憶する。

【０１５３】ステップ２１０６での繰り返しを終了するとステップ２１０８に進み、前記ステップ２１０７で対応付けを行った構造化文字位置情報群について連接判定処理を行い、連続した文字列として前記指定文字列を構成する構造化文字位置情報のグループのみを抽出し、該抽出した各グループ中で前記指定文字列の先頭に位置する部分文字列に対応する構造化文字位置情報のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【０１５４】ステップ２１０３から分岐したすべての処理を終えるとステップ２１０９に進み、前記ヒット位置集合中に含まれる構造化文字位置情報群を、同一の文書識別子を持つものからなるグループにまとめて前記検索結果データに登録する。

【０１５５】ここで、前記ステップ２１０８、すなわち文字列インデックス検索プログラム１７０２の処理過程における連接判定処理について、図２２を用いてさらに詳しく説明する。

【０１５６】図２２において、２２０１は、文字列インデックスの一例（部分）を表わしている。２２０１に示すデータを保持している文字列インデックス上で、図２０の２００３に示す展開済み検索条件データが示す条件に従って検索を行うと、前記ステップ２１０７に示したとおり、まず部分文字列“ガー”および“ード”に対応

する構造化文字位置情報群の中から、文脈識別子がＣ３またはＣ９であるもののみが抽出される。該抽出された構造化文字位置情報群を、部分文字列に対応付けたデータを２２０２に示す。連接判定処理は該データの上で行われる。

【０１５７】前記ステップ２１０８に示す連接判定処理では、抽出された構造化文字位置情報群の中に、全体として前記指定文字列を構成する一連の構造化文字位置情報の組み合わせが存在するかどうかを判定する。ここで、このような組み合わせは次に示す条件を満たさなければならない。

【０１５８】（１）構造化文字位置情報群の間で、文書識別子がすべて一致している。

【０１５９】（２）構造化文字位置情報群の間で、文脈識別子がすべて一致している。

【０１６０】（３）構造化文字位置情報を、その文字位置の値が小さいものから順に並べ、それらの文字位置に従って対応する部分文字列群を配置すると、全体として指定文字列に等しい文字列が得られる。

【０１６１】ここで、２２０２に示す事例には、全体として指定文字列“ガード”を構成する組み合わせが一例含まれている。

【０１６２】上記の条件を満たす構造化文字位置情報の組み合わせが見つかると、該組み合わせ中に含まれる構造化文字位置情報群のうち、文字位置の値が最小であるものを代表として選び、前記ヒット位置集合に登録する。

【０１６３】次に、図２３は、個々の検索処理の結果として生成される検索結果データのデータ構造を示す図である。本図に示すとおり、検索結果データは、ヒット位置集合中に含まれる文字位置情報群を文書識別子ごとにグループ化し、それらのグループを要素とするリストを作り、さらに検出文書の総数を示す情報を付加した構造を持つ。検索結果データは、検索結果データの集合中でその検索結果データを一意に識別する検索結果識別子と対応付けられたうえで、検索結果データ格納領域１７０４に格納される。

【０１６４】次に、図１８におけるステップ１８０９、すなわち検索結果問い合わせリクエストの内容に対応して検索結果を要求元クライアントに転送する処理について、図２４を用いてさらに詳しく説明する。図２４は、前記ステップ１８０９の詳細な処理手順を示すＰＡＤ図である。

【０１６５】ここで、検索結果問い合わせリクエストの本体は、検索結果識別子指定、問い合わせ種別指定および文書識別子指定の三部分からなる。問い合わせの種別によっては、文書識別子指定をもたない場合もある。

【０１６６】図２４に示すとおり、前記ステップ１８０９に対応する処理では、まずステップ２４０１において、問い合わせリクエスト中で指定されている検索結果

識別子に対応する検索結果データを検索し、該検索結果データを検索結果データ格納領域１７０４から読み込む。

【０１６７】次に、ステップ２４０２では、問い合わせ種別を判定し、該問い合わせ種別が検出文書数問い合わせであった場合にはステップ２４０３、該問い合わせ種別が文書識別子問い合わせであった場合にはステップ２４０４、該問い合わせ種別が文字位置情報問い合わせであった場合にはステップ２４０５に分岐する。

【０１６８】ステップ２４０３では、前記ステップ２４０１で読み込んだ検索結果データから検出文書数を抽出し、該検出文書数の値を要求元の検索クライアントに転送して処理を終了する。

【０１６９】ステップ２４０４では、前記ステップ２４０１で読み込んだ検索結果データ中に含まれるすべての文書識別子からなる集合を求め、該集合を要求元の検索クライアントに転送して処理を終了する。

【０１７０】ステップ２４０５では、前記ステップ２４０１で読み込んだ検索結果データから、問い合わせ中で指定された文書識別子に対応する構造化文字位置情報のリストを抽出し、該リストを要求元の検索クライアントに転送して処理を終了する。以上が、本実施例における文書検索サーバ１０２の説明である。

【０１７１】以下、本発明の第一の実施例における文書検索クライアント、すなわち図１の１０３および１０４について説明する。

【０１７２】図２５は、１０３および１０４に共通する、本実施例における文書検索クライアントの構成を示す図である。

【０１７３】図２５に示す文書検索クライアントは、ディスプレイ２０１、キーボード２０２、中央処理装置（ＣＰＵ）２０３、通信制御装置２０６、主メモリ２０７、磁気ディスク装置２０８、システムバス２０９から構成される。

【０１７４】ディスプレイ２０１は、ユーザが対話的に検索条件を入力するための画面や検索結果等を表示するために使用する。キーボード２０２は、検索条件、検索処理の実行等を指示するコマンドを入力するために使用する。中央処理装置２０３は、本クライアントを構成する各種プログラムを実行する。通信制御装置２０６は、ネットワーク１０５を介して前記文書検索サーバ１０２と通信し、リクエストおよびデータの交換を行うために使用する。主メモリ２０７は、本クライアントによる処理を実行するための各種プログラムおよび一時的なデータを保持するために使用する。磁気ディスク装置２０８は、検索結果として得られた文書およびその他のデータを格納するために使用する。システムバス２０９は、これらの各種装置を接続するために使用する。

【０１７５】主メモリ２０７中には、検索条件入力プログラム２５０１、検索結果表示プログラム２５０２、クライアント制御プログラム２５０３およびシステムプログラム２１５が格納され、またワークエリア２１６が保持される。磁気ディスク装置２０８中には、解析済み文書データ格納領域２１７および検索結果データ格納領域１７０４が確保される。

【０１７６】検索条件入力プログラム２５０１は、ユーザと対話しつつ検索条件の入力および解釈を行う。検索結果表示プログラム２５０２は、文書検索サーバ１０２から受け取った検索結果の表示を行う。クライアント制御プログラム２５０３は、検索条件入力プログラム２５０１および検索結果表示プログラム２５０２の起動および実行制御を行うとともに、ネットワーク１０５を介して、文書検索サーバ１０２との間でリクエストおよびデータの交換を行う。システムプログラム２１５は、周辺装置との間のデータの入出力など、コンピュータ上で本クライアントを構成する各プログラムを実行するための基本機能を提供する。ワークエリア２１６は、プログラムの実行時に一時的に必要となるデータを記憶するために用いられる。

【０１７７】なお、本実施例では文書検索サーバ１０２との間でデータを転送するためにネットワーク１０５を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。また、１個以上の検索クライアントを文書検索サーバ１０２と同一のコンピュータ上で実行し、これらの間でデータ転送を行わない構成をとることもできる。本クライアントにプリンタを接続し、検索結果を印刷できるよう構成することもできる。

【０１７８】図２６は、本発明の第一の実施例における検索クライアントの動作手順を示すＰＡＤ図である。キーボード２０２からのクライアント起動コマンド等によりクライアント制御プログラム２５０３が起動されると、本プログラムはユーザから文書検索を指示するコマンドを受け取ってはその処理を行うループに入る（ステップ２６０１）。このループは、キーボード２０２からクライアントの停止を指示するコマンドが入力されるまで継続する。

【０１７９】ステップ２６０１のループは、ステップ２６０２からステップ２６０５までに示す処理を繰り返す。

【０１８０】ステップ２６０２では、検索条件入力プログラム２５０１を実行し、ユーザとの対話により検索条件を入力し、該検索条件を、文書検索サーバ１０２が解釈可能な文書検索リクエストに変換する。ステップ２６０３では、前記文書検索リクエストを、ネットワーク１０５を介して文書検索サーバ１０２に送信する。ステップ２６０４では、文書検索サーバ１０２から前記文書検索リクエストへの返送として検索結果識別子が返されるのを待ち、該識別子を受信する。

35

【０１８１】ステップ２６０５では、前記検索結果識別子を入力として検索結果表示プログラム２５０２を実行し、ユーザと対話しつつ検索結果データの問い合わせおよび画面表示を行う。

【０１８２】図２７は、前記図２６のステップ２６０２において実行する検索条件入力プログラム２５０１の詳細な処理手順を示すＰＡＤ図である。検索条件入力プログラム２５０１は、クライアント制御プログラム２５０３から起動されると、まず検索条件をユーザが対話的に指定するための画面をディスプレイ２０１に表示する（ステップ２７０１）。

【０１８３】次に、ステップ２７０２において、前記画面上でユーザが指示した検索条件を読み込む。

【０１８４】次に、ステップ２７０３において、前記ステップ２７０２において読み込んだ検索条件を、文書検索サーバ１０２が直接解釈可能な文書検索リクエストの形に変換する。

【０１８５】図２８は、前記図２６のステップ２６０５において実行する検索結果表示プログラム２５０２の詳細な処理手順を示すＰＡＤ図である。検索結果表示プログラム２５０２は、クライアント制御プログラム２５０３から前記検索結果識別子を入力として起動されると、ただちにステップ２８０１のループに入る。該ループは、ユーザから検索結果表示の終了を指示されるまで、ステップ２８０２からステップ２８１５までに示す処理を、繰り返し実行する。

【０１８６】前記ステップ２８０１のループ内では、まずステップ２８０２において、検索結果の表示とユーザからの指示入力のために用いる画面をディスプレイ２０１に表示する。次に、ステップ２８０３において、前記画面上でユーザが指定した指示内容を読み込む。

【０１８７】次に、ステップ２８０４において、前記ユーザ指示の種別を判定し、その種別に対応した分岐を行う。すなわち、該指示が検出文書数の表示を求めるものであった場合にはステップ２８０５およびステップ２８０６に示す処理に分岐し、該指示が検出文書群の文書識別子リストの表示を求めるものであった場合にはステップ２８０７およびステップ２８０８に示す処理に分岐し、該指示が文書内容の表示を求めるものであった場合にはステップ２８０９からステップ２８１５までに示す処理に分岐する。各分岐先の処理が終了するとステップ２８０２に戻り、再び前記ループを再開する。

【０１８８】ここで、ステップ２８０５では、検出文書数を問い合わせるための検出文書数問い合わせリクエストを生成し、該リクエストを文書検索サーバ１０２に送信する。次に、ステップ２８０６では、前記リクエストに対応して文書検索サーバ１０２から転送されてきた検出文書数を受信し、該数値をディスプレイ２０１に表示する。

【０１８９】ステップ２８０７では、検出文書群の文書識別子リストを問い合わせるための文書識別子問い合わせリクエストを生成し、該リクエストを文書検索サーバ１０２に送信する。次に、ステップ２８０８では、前記リクエストに対応して文書検索サーバ１０２から転送されてきた文書識別子の集合を受信し、該集合に含まれる文書識別子群をディスプレイ２０１にリスト表示する。

36

【０１９０】ステップ２８０９では、表示すべき文書を特定する文書識別子を入力する。次に、ステップ２８１０では、該識別子が識別する文書の解析済み文書データを得るための文書転送リクエストを生成し、該リクエストを文書検索サーバ１０２に送信する。次に、ステップ２８１１では、前記リクエストに対応して文書検索サーバ１０２から転送されてきた解析済み文書データを受信し、該データを解析済み文書データ格納領域２１７に格納する。

【０１９１】次に、ステップ２８１２では、前記解析済み文書データ中において、検索条件中に指定した指定文字列が検出された位置を問い合わせるための文字位置情報問い合わせリクエストを生成し、該リクエストを文書検索サーバ１０２に送信する。次に、ステップ２８１３では、前記リクエストに対応して文書検索サーバ１０２から転送されてきた構造化文字位置情報のリストを受信し、該リストを検索結果データ格納領域１７０４に格納する。

【０１９２】次に、ステップ２８１４では、前記ステップ２８１１で受信した解析済み文書データおよび前記ステップ２８１３で受信した構造化文字位置情報リストを参照し、文書検索時における指定文字列検出部分を反転表示するためのデータ加工処理を行う。次に、ステップ２８１５では、前記反転処理済みの解析済み文書データを、書式化してディスプレイ２０１上に表示する。

【０１９３】以上が、本発明の第一の実施例における検索クライアント１０３および１０４の動作手順の説明である。

【０１９４】（第二の実施例）以下、本発明を適用した第二の実施例について、図面を用いて説明する。

【０１９５】図２９は、本実施例における文書登録サブシステム１０１の構成を示す図である。

【０１９６】図２９に示す文書登録サブシステム１０１は、そのハードウェア構成に関しては、図２に示す第一の実施例の場合と変わらない。ただし、主メモリ２０７中には、第一の実施例において保持するプログラム群に加えて、逆順構造インデックス作成プログラム２９０１を保持する。また、磁気ディスク装置２０８中には、第一の実施例において確保する領域群に加えて、逆順構造インデックス格納領域２９０２が確保される。逆順構造インデックス作成プログラム２９０１は、登録対象文書の解析済み文書データが持つ論理構造に関する情報を、逆順構造インデックス格納領域２９０２に格納されている逆順構造インデックスに登録して逆順構造インデック

スを更新する。

【０１９７】本実施例において、文書登録制御プログラム２１４は、文書構造解析プログラム２１０、構造インデックス作成プログラム２１１、逆順構造インデックス作成プログラム２９０１、構造化全文データ生成プログラム２１２および文字列インデックス作成プログラム２１３の起動および実行制御を行うとともに、これらのプログラムによって生成された解析済み文書データ、構造インデックス、逆順構造インデックスおよび文字列インデックスをネットワーク１０５を介して文書検索サーバ１０２に転送する。

【０１９８】なお、本実施例ではフロッピーディスク２０５に格納された登録対象文書を入力として読み込む構成としたが、光磁気ディスク、追記型光ディスク等、他種の可搬型媒体から読み込む構成をとることもでき、ネットワーク１０５を介して転送されてくる文書を入力とする構成をとることもできる。さらに、本実施例では生成された解析済み文書データ、構造インデックス、逆順構造インデックスおよび文字列インデックスを文書検索サーバ１０２に転送するためにネットワーク１０５を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。あるいは、文書登録サブシステム１０１と文書検索サーバ１０２を１台のコンピュータ上に実装し、データ転送を行わない構成をとることもできる。

【０１９９】図３０は、本発明の第二の実施例における文書登録処理の概略手順を示すＰＡＤ図である。本図に示す処理手順は、図３に示す第一の実施例における処理手順とほぼ同様であるが、図３におけるステップ３０５の直後にステップ３００１が追加されている点と、ステップ３０８の代わりにステップ３００２を実行する点が異なる。

【０２００】ここで、ステップ３００１では、ステップ３０４で生成した解析済み文書データを入力として逆順構造インデックス作成プログラム２９０１を実行する。逆順構造インデックス作成プログラム２９０１は、まず逆順構造インデックス格納領域２９０２から現時点での逆順構造インデックスを読み出し、与えられた解析済み文書データの持つ構造情報をこの逆順構造インデックスに登録し、更新された逆順構造インデックスを再び逆順構造インデックス格納領域２９０２に格納する。

【０２０１】ステップ３００２では、解析済み文書データ格納領域２１７に格納されたすべての解析済み文書データ、構造インデックス格納領域２１８に格納された構造インデックス、逆順構造インデックス格納領域２９０２に格納された逆順構造インデックス、および文字列インデックス格納領域２２０に格納された文字列インデックスを、ネットワーク１０５を介して文書検索サーバ１０２に転送する。

【０２０２】図３１は、図３０におけるステップ３００１の詳細、すなわち本実施例における逆順構造インデックス作成プログラム２９０１の処理手順を示すＰＡＤ図である。

【０２０３】逆順構造インデックス作成プログラム２９０１は、まずステップ３１０１において、既存の逆順構造インデックスが逆順構造インデックス格納領域２９０２中に存在しているかどうかを判定する。該領域中に逆順構造インデックスが存在しない場合にはステップ３１０２に進み、初期状態（空）の逆順構造インデックスを生成する。既存の逆順構造インデックスを検出した場合にはステップ３１０３に進み、該逆順構造インデックスを読み込む。

【０２０４】次に、ステップ３１０４において、登録対象文書の解析済み文書データを読み込む。

【０２０５】次に、ステップ３１０５において、前記解析済み文書データの木構造を構成するすべてのノード（要素および文字列データ）を対象として、ステップ３１０６からステップ３１０９に示す処理を繰り返す。

【０２０６】ステップ３１０６では、解析済み文書データ中で現在着目しているノードについて、逆順構造インデックス中に、該ノードに対応するメタノード（メタ要素またはメタ文字列データ）が存在するかどうかを判定する。対応メタノードが存在しない場合ステップ３１０７に進み、該ノードに対応するメタノードを生成して逆順構造インデックスに登録し、さらに該登録したメタノードに対して、該メタノードを逆順構造インデックス中で一意に識別する番号である逆順文脈識別子を割り当てる（ステップ３１０８）。

【０２０７】ステップ３１０９では、解析済み文書データ中で現在着目しているノードと、逆順構造インデックス中で該ノードに対応するメタノードを識別する逆順文脈識別子との対応関係を、解析済み文書データに付加して該解析済み文書データを更新する。

【０２０８】ステップ３１０５以下の繰り返しを終了するとステップ３１１０に進み、前記更新された解析済み文書データを出力して解析済み文書データ格納領域２１７に格納する。次に、ステップ３１１１では、前記更新された逆順構造インデックスを出力して逆順構造インデックス格納領域２９０２に格納し、処理を終了する。

【０２０９】以上に述べたとおり、逆順構造インデックス作成プログラム２９０１の処理手順は図９に示す構造インデックス作成プログラム２１１の処理手順とほぼ対応している。しかし、ステップ３１０５の繰り返しにおいて解析済み文書の木構造を辿る順序が構造インデックス作成プログラム２１１の場合とは異なっており、その結果構築される逆順構造インデックスの木構造も構造インデックスの木構造とは異なったものとなる。

【０２１０】ここで、前記ステップ３１０５において解析済み文書データの木構造を構成するすべてのノードを

対象とする繰り返しを行う際に、該木構造を辿り、個々のノードを処理していく順序を図３２を用いて説明する。図３２では、楕円形で要素ノード、矩形で文字列ノードを表現し、あるノードが複数の下位ノードを持つ場合には、各下位ノードをその出現順に左から右に並べて表現している。また、各ノードに表示された数字がその処理順序を示している。

【０２１１】図３２に示すとおり、ステップ３１０５においてノード群を処理していく順序は、木構造の根に位置するノードから出発し、ある特定のノードとその下位ノード群を処理する際には、まず該ノードの処理を行い、次に該ノードの下位ノード群を、その出現順の逆順に従って処理していく順序となる。

【０２１２】次に、前記ステップ３１０６における処理、すなわち解析済み文書データ中で現在着目しているノードについて、逆順構造インデックス中に、該ノードに対応するメタノードが存在するかどうかを判定する処理について、図３３を用いて説明する。図３３は、図の左側に示す解析済み文書データの木構造を構成するノード群と、右側に示す逆順構造インデックスの木構造を構成するノード（メタノード）群との対応関係を示す図である。

【０２１３】ここで、本実施例では、解析済み文書の木構造中におけるあるノードの逆順木構造アドレスと、逆順構造インデックスの木構造中におけるあるメタノードの逆順木構造アドレスとが等しい場合に、該ノードと該メタノードとは対応するものと定義する。ここでいう逆順木構造アドレスとは、木構造の根から出発して上位ノードから下位ノードへと辿り、特定のノードに至るまでの道筋を、該道筋上に存在する各ノードの種別（要素か文字列データか、要素の場合、どの要素型に属するか）と、それらのノードが種別の等しい兄弟ノード群中で後ろから何番目に現れたかを示す番号との組み合わせによって表現するアドレスのことである（通常の木構造アドレスと区別するため、逆順木構造アドレスでは前記番号を負の整数として表記する）。

【０２１４】例えば、図３３に示す解析済み文書データ中のノード群のうち、ノード３３０１は上位ノードをもたず、兄弟ノード中では最後の“論文”要素ノードであるから、その逆順木構造アドレスは“/論文[-1]”と表記することができる。同様に、ノード３３０２はノード３３０１の下位ノードであり、兄弟ノード中では最後の“章”要素ノードであるから、その逆順木構造アドレスは“/論文[-1]/章[-1]”と表記することができる。また、ノード３３０３はノード３３０２の下位ノードであり、兄弟ノード中では後ろから２番目の“節”要素ノードであるから、その逆順木構造アドレスは“/論文[-1]/章[-1]/節[-2]”と表記することができる。また、ノード３３０４はノード３３０３の下位ノードであり、兄弟ノード中では最後の“段落”要素ノードであるから、その逆順木構造アドレスは“/論文[-1]/章[-1]/節[-2]/段落[-1]”と表記することができる。

【０２１５】同様にして、図３３右側の構造インデックスの木構造を構成する各メタノードについてその逆順木構造アドレスを求めてみると、メタノード３３０５の逆順木構造アドレスは“/論文[-1]”となり、ノード３３０１の逆順木構造アドレスと等しくなる。同様に、メタノード３３０６の逆順木構造アドレスは“/論文[-1]/章[-1]”となってノード３３０２の逆順木構造アドレスと等しく、メタノード３３０７の逆順木構造アドレスは“/論文[-1]/章[-1]/節[-2]”となってノード３３０３の逆順木構造アドレスと等しくなる。よって、前記ステップ３１０６において、ノード３３０１はメタノード３３０５、ノード３３０２はメタノード３３０６、ノード３３０３はメタノード３３０７とそれぞれ対応するものと判定される。なお、図３３の構造インデックス中にはノード３３０４と等しい逆順木構造アドレスを持つメタノードはないので、ノード３３０４と対応すメタノードは逆順構造インデックス中に存在しないものと判定され、前記ステップ３１０７において新たなメタノードが生成され、該メタノードが構造インデックスに登録されることになる。

【０２１６】前記ステップ３１０７において、あるノードに対応する新たなメタノードを登録する際には、該ノードの上位ノードに対応するメタノードの持つ下位メタノード群の先頭に、該ノードに対応する種別のメタノードを追加する。図３３のノード３３０４に対応するメタノードを登録する場合には、ノード３３０４の上位ノードであるノード３３０３に対応するメタノード３３０７の下位に、要素型“段落”のメタノードが追加され、該メタノードは兄弟メタノード群の先頭に置かれることになる。

【０２１７】次に、複数の解析済み文書データを順次重ね合わせることにより逆順構造インデックスを生成していく過程について、図３４を用いて説明する。図３４において、３４０１、３４０３および３４０５は、それぞれ登録対象文書の解析済み文書データを表わしている。これらの解析済み文書データの構造を既存の逆順構造インデックス上に順次重ね合わせることにより、逆順構造インデックスが形成されていく。

【０２１８】まず最初に文書１の解析済み文書データ３４０１が入力されると、最初の段階では逆順構造インデックスは初期状態（空）であるため、該解析済みデータと等価な木構造が生成されてそのまま逆順構造インデックスに登録され、逆順構造インデックスは３４０２に示す状態となる。新たに生成されたメタ要素には－Ｅ１から－Ｅ５までの文脈識別子、新たに生成されたメタ文字列データには－Ｃ１から－Ｃ３までの文脈識別子が割り当てられる。

【０２１９】次に文書２の解析済み文書データ３４０３

41

が入力されると、既存の逆順構造インデックス（３４０２）と構造が重複する部分については何も行わず、３４０２上に対応する部分がなかった部分構造（図中の斜線部分）だけが新たに登録される。新たに生成されたメタ要素には文脈識別子－Ｅ６および－Ｅ７、新たに生成されたメタ文字列データには文脈識別子－Ｃ４が割り当てられる。

【０２２０】次に文書３の解析済み文書データ３４０５が入力されると、既存の逆順構造インデックス（３４０４）と構造が重複する部分については何も行わず、３４０４上に対応する部分がなかった部分構造（図中の斜線部分）だけが新たに登録される。新たに生成されたメタ要素には文脈識別子－Ｅ８、－Ｅ９および－Ｅ１０、新たに生成されたメタ文字列データには文脈識別子－Ｃ５および－Ｃ６が割り当てられる。このようにして、３個の文書が登録された段階で、逆順構造インデックスは３４０６に示す状態となる。

【０２２１】図３５は、図３０におけるステップ３０６の詳細、すなわち本実施例における構造化全文データ生成プログラム２１２の処理手順を示すＰＡＤ図である。図３５に示すとおり、本実施例における構造化全文データ生成プログラム２１２の処理手順は、図１３に示す前記第一の実施例の場合の処理手順とほぼ同一である。ただし、本実施例の場合、図１３におけるステップ１３０５の代わりにステップ３５０１を行う点が異なっている。

【０２２２】ここで、ステップ３５０１では、解析済み文書データから、現在着目している文字列データノードに対応する文脈識別子および逆順文脈識別子を求め、該文脈識別子および該逆順文脈識別子を構造化全文データ格納領域２１９に出力する。

【０２２３】図３６に、本実施例において構造化全文データ生成プログラム２１２が出力する構造化全文データのファイル形式を示す。図３６は、図５に示したＳＧＭＬ文書を入力として構造化全文データを生成した場合について例示している。図３６に示すとおり、本実施例における構造化全文データのデータファイルは、先頭に文書識別子を記述し、その後ろに、文脈識別子、逆順文脈識別子およびそれらに対応する内容文字列からなる三つ組みを、文書中に存在する文字列データの数だけ繰り返した構造をとる。

【０２２４】本実施例における文字列インデックスは、前記第一の実施例の場合と同じく、図１５に示す処理手順に従って作成される。図３７に、本実施例における文字列インデックスのデータ構造を示す。図３７は、文字列インデックス作成プログラム２１３を用いて図３６に示す構造化全文データを処理し、文字列インデックスに該構造化全文データに含まれる部分文字列群の登録を終えた段階において、該文字列インデックスのデータ構造の一部（内容文字列“変換処理の一例”に関連する部分）を図示したものである。

42

【０２２５】図３７に示すとおり、本実施例における文字列インデックスでは、個々の構造化文字位置情報中に、前記第一の実施例の場合に保持する情報に加えて、逆順文脈識別子をも保持する。ただし、図３７では、前記第一の実施例における図１６と同様、前記内容文字列の末尾“例”に対応する文字ノードおよび構造化文字位置情報は省略している。また、前記内容文字列の直前に位置する文字の文字位置を“Ｘ”として、相対的に文字位置を表現している。

【０２２６】以上が、本実施例における文書登録サブシステム１０１の説明である。

【０２２７】以下、本発明の第二の実施例における文書検索サーバ、すなわち図１の１０２について説明する。

【０２２８】図３８は、本実施例における文書検索サーバ１０２の構成を示す図である。図３８に示すとおり、本実施例における文書検索サーバ１０２は、前記第一の実施例の場合の構成要素群に加えて、磁気ディスク装置２０８中に、逆順構造インデックス格納領域２９０２を保持する。

【０２２９】なお、本実施例の場合も、文書登録サブシステム１０１および検索クライアントとの間でデータを転送するためにネットワーク１０５を使用する代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。また、文書登録サブシステム１０１と文書検索サーバ１０２を１台のコンピュータ上に実装し、これらの間でデータ転送を行わない構成をとることもできる。また、１個以上の検索クライアントを文書検索サーバ１０２と同一のコンピュータ上で実行し、これらの間でデータ転送を行わない構成をとることもできる。

【０２３０】図３９は、本発明の第二の実施例における文書検索処理の概略手順を示すＰＡＤ図である。図３９に示すとおり、本実施例における文書検索処理の手順は、図１８に示す前記第一の実施例の場合とほぼ同一であるが、前記第一の実施例におけるステップ１８０５の代わりにステップ３９０１を実行する点が異なっている。ステップ３９０１では、文書登録サブシステム１０１から、新規に登録された文書群の内容を反映して更新された構造インデックス、逆順構造インデックスおよび文字列インデックスを受信し、これらをそれぞれ構造インデックス格納領域２１８、逆順構造インデックス格納領域２９０２および文字列インデックス格納領域２２０に格納する。

【０２３１】図４０は、図３９におけるステップ１８０６の詳細、すなわち本実施例における検索条件解析プログラム１７０１の処理手順を示すＰＡＤ図である。

【０２３２】本実施例の場合、検索条件解析プログラム１７０１は、文書検索リクエスト中で指定されている検索条件を入力として起動されると、まずステップ４００

１において、該検索条件中に含まれている構造条件について判定する。ここで、該検索条件中に正順の（すなわち前記第一の実施例におけるものと同様な）構造条件が含まれている場合には、ステップ１９０２およびステップ１９０３からなる処理に分岐する。ここで、ステップ１９０２および１９０３での処理は、前記第一の実施例の場合と同一である。該検索条件中に逆順の構造条件が含まれている場合には、ステップ４００２およびステップ４００３からなる処理に分岐する。該検索条件中に構造条件が含まれていなかった場合には何もせずにステップ１９０４に進む。

【０２３３】ステップ４００２では、逆順構造インデックス格納領域２９０２から逆順構造インデックスを読み込む。次に、ステップ４００３では、該逆順構造インデックスを参照して、前記構造条件を満たす構造内に含まれるすべての文字列データの逆順文脈識別子の集合を求める。以下、該集合を逆順文脈識別子集合と呼ぶ。

【０２３４】ステップ１９０４およびそこから分岐するステップ１９０５およびステップ１９０６における処理は、前記第一の実施例の場合と同一であり、これらの処理を終了した後、ステップ４００４に進む。

【０２３５】ステップ４００４では、ステップ１９０３で求めた文脈識別子集合、前記ステップ４００３で求めた逆順文脈識別子集合、前記検索条件中に含まれていた指定文字列、および前記ステップ１９０５またはステップ１０９６で生成した部分文字列リストから構成される展開済み部分文字列データを生成して処理を終了する。

【０２３６】ここで、図４１は、本実施例での検索条件解析プログラム１７０１の処理過程における、展開済み解析条件データの生成例を示す図である。

【０２３７】図４１において、４１０１は、文書検索リクエスト中で指定された検索条件の一例である。検索条件４１０１は、構造条件指定“章/段落[-1]”と、文字列条件指定“ガード”とから構成されている。前記検索条件は、“章”要素の直接の下位にある最後の“段落”要素内に、文字列“ガード”が出現するケースを検索すべきことを指定している。

【０２３８】ここで、前記検索条件中で指定されている構造条件は、構造を後ろ側からに辿って条件を指定する逆順の構造条件であるから、逆順構造インデックスの内容が４１０２に示すとおりであったとすると、前記ステップ４００３においてこの構造インデックスを参照することにより、前記構造条件指定を満たす“段落”要素の逆順文脈識別子は－Ｅ３および－Ｅ１２であることがわかる。従って、これらの段落の下位にある文字列データ、すなわち逆順文脈識別子が－Ｃ１または－Ｃ７である文字列データ内に、文字列“ガード”が出現するケースを検索すればよいことがわかる。ただし、検索に用いる文字列インデックスには、長さ２の部分文字列についてのみその出現位置が登録されているので、３文字からなる前記指定文字列を直接検索することはできない。そこで、ステップ１９０５において、前記指定文字列を分解して長さ２の部分文字列からなるリストを生成する。前記のとおり指定文字列が“ガード”だった場合、抽出される部分文字列は“ガー”および“ード”となる。

【０２３９】この結果、前記ステップ４００４において、４１０３に示す展開済み検索条件データ、すなわち文脈識別子集合が空集合、逆順文脈識別子集合が｜－Ｃ１，－Ｃ７｜、指定文字列が“ガード”、部分文字列リストが｜“ガー”，“ード”｜であるデータが生成される。

【０２４０】図４２は、図３９におけるステップ１８０７の詳細、すなわち本実施例における文字列インデックス検索プログラム１７０２の処理手順を示すＰＡＤ図である。

【０２４１】文字列インデックス検索プログラム１７０２は、前記検索条件解析プログラム１７０１が生成した展開済み検索条件データを入力として起動される。図４２に示すとおり、本プログラムの処理手順は図２１に示す第一の実施例の場合の処理手順とほぼ同様であるが、前記図２１におけるステップ２１０４、ステップ２１０５およびステップ２１０７に代わって、それぞれステップ４２０１、ステップ４２０２およびステップ４２０３が実行される。

【０２４２】ステップ４２０１では、文字列インデックス中で前記展開済み検索条件データ中における指定文字列を検索し、該文字列に対応している構造化文字位置の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子、または前記展開済み検索条件データ中の逆順文脈識別子集合に含まれているいずれかの逆順文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【０２４３】ステップ４２０２では、文字列インデックス中で前記指定文字列を検索し、該文字列の末尾に対応する文字ノードより前方に存在するすべての構造化文字位置情報の集合を求め、次に該集合の中から、前記展開済み検索条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子、または前記展開済み検索条件データ中の逆順文脈識別子集合に含まれているいずれかの逆順文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群からなるヒット位置集合を生成する。

【０２４４】ステップ４２０３では、前記展開済み検索条件データ中における部分文字列リスト中に含まれている部分文字列群のうち、ステップ２１０６の繰り返し中で現在着目している部分文字列を文字列インデックス中で検索し、該文字列に対応している構造化文字位置情報の集合を求め、次に該集合の中から、前記展開済み検索

条件データ中の文脈識別子集合に含まれているいずれかの文脈識別子、または前記展開済み検索条件データ中の逆順文脈識別子集合に含まれているいずれかの逆順文脈識別子を持つ構造化文字位置情報群のみを抽出し、該抽出した構造化文字位置情報群を前記部分文字列に対応付けて記憶する。

【０２４５】図４２におけるステップ２１０８、すなわち文字列インデックス検索プログラム１７０２の処理過程における連接判定処理は、図２２に示す前記第一の実施例の場合と同様となる。ただし、検索条件中の構造条件が逆順の構造条件だった場合には、連接判定の際、文脈識別子の一致ではなく逆順文脈識別子の一致を判定することになる。

【０２４６】以上説明したように、本実施例に示した構成によれば、前記第一の実施例において検索が可能となる各種構造条件に加えて、「論文中で最後の章の中で、ある特定の文字列を検索する」、あるいは「後ろから２番目の参考文献の中で、ある特定の文字列を検索する」のように、文書の論理構造を逆順に（後ろから）辿った構造条件を指定した検索も可能となる。

【０２４７】以上が、本発明の第二の実施例の説明である。

【０２４８】（第三の実施例）以下、本発明を適用した第三の実施例について、図面を用いて説明する。

【０２４９】本実施例は、システムの構成および各プログラムの処理手順のいずれについても前記第一の実施例と同一であるが、文書木構造中のノードと構造インデックス中のメタノードとの対応のさせ方が異なり、その結果、同一の文書群を入力とした場合でも構造インデックスの構造および文脈インデックスの割り当てが前記第一の実施例とは異なってくる。

【０２５０】ここで、本実施例における、文書木構造中のノードと構造インデックス中のメタノードとの対応関係を、図４３を用いて説明する。図４３は、図の左側に示す解析済み文書データの木構造を構成するノード群と、右側に示す構造インデックスの木構造を構成するノード（メタノード）群との対応関係を示している。

【０２５１】本実施例においても、解析済み文書の木構造中におけるあるノードの木構造アドレスと、構造インデックスの木構造中におけるあるメタノードの木構造アドレスとが等しい場合に、該ノードと該メタノードとは対応するものと定義する。ただし、本実施例では、前記第一の実施例とは異なり、共通の上位ノードを持つ同種の兄弟ノード間での出現順位を考える際に、先頭のノードと２番目のノードとの区別は行うが、２番目のノードとそれ以降に現れたノードとの区別は行わない。すなわち、木構造アドレス中で出現順位を表わす番号は常に[1]または[2]のいずれかをとり、[3]以上の値をとることはない。

【０２５２】例えば、図４３に示す解析済み文書データ

中のノード群のうち、ノード４３０１は上位ノードをもたず、兄弟ノード中では最初の“論文”要素ノードであるから、その木構造アドレスは“/論文[1]”と表記することができる。同様に、ノード４３０２はノード４３０１の下位ノードであり、兄弟ノード中では最初の“章”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]”となる。これに対して、ノード４３０３はノード４３０２の下位ノードであり、兄弟ノード中では４番目の“節”要素ノードであるが、前記規則から、その木構造アドレスは“/論文[1]/章[1]/節[2]”となる。また、ノード４３０４はノード４３０３の下位ノードであり、兄弟ノード中では２番目の“段落”要素ノードであるから、その木構造アドレスは“/論文[1]/章[1]/節[2]/段落[2]”となる。

【０２５３】同様にして、図４３右側の構造インデックスの木構造を構成する各メタノードについてその木構造アドレスを求めてみると、メタノード４３０５の木構造アドレスは“/論文[1]”となり、ノード４３０１の木構造アドレスと等しくなる。同様に、メタノード４３０６の木構造アドレスは“/論文[1]/章[1]”となってノード４３０２の木構造アドレスと等しく、メタノード４３０７の木構造アドレスは“/論文[1]/章[1]/節[2]”となってノード４３０３の木構造アドレスと等しくなる。よって、ノード４３０１はメタノード４３０５、ノード４３０２はメタノード４３０６、ノード４３０３はメタノード４３０７とそれぞれ対応するものと判定される。なお、図４３の構造インデックス中にはノード４３０４と等しい木構造アドレスを持つメタノードはないので、ノード４３０４と対応するメタノードは構造インデックス中に存在しないものとみなされる。

【０２５４】以上説明したように、上記規則の結果、本実施例において形成される構造インデックスでは、同一のメタノードの下位に、同種のメタノードが３個以上付加されることはなく、文書木構造中で同種のノードが３個以上連続している場合、２番目以降のノードには同一の文脈識別子が割り当てられる。よって、本実施例による文書検索方法では、構造条件中で任意の出現順位を指定することはできず、同種要素中で先頭の要素か、２番目以降の要素かを区別できるだけとなる。その代わり、本実施例では構造インデックスのデータ構造が前記第一の実施例の場合と比べて簡略になり、構造インデックス格納領域２１８として必要となる容量を削減することができる。

【０２５５】なお、本実施例におけるノードとメタノードとの対応付けを採用した場合においても、前記第二の実施例同様に正順・逆順二つの構造インデックスを用意することにより、出現順を後ろから辿った構造指定をも可能とすることができる。

【０２５６】以上が、本発明の第三の実施例の説明である。

【０２５７】（第四の実施例）以下、本発明を適用した第四の実施例について、図面を用いて説明する。

【０２５８】図４４は、本実施例における文書登録サブシステム１０１の構成を示す図である。

【０２５９】図４４に示す文書登録サブシステム１０１は、そのハードウェア構成および磁気ディスク２０８中の格納領域の構成に関しては、図２に示す第一の実施例の場合と変わらない。ただし、主メモリ２０７中には、第一の実施例において保持するプログラム群に加えて、文書構造正規化プログラム４４０１を保持する。

【０２６０】本実施例において、文書登録制御プログラム２１４は、文書構造解析プログラム２１０、文書構造正規化プログラム４４０１、構造インデックス作成プログラム２１１、構造化全文データ生成プログラム２１２および文字列インデックス作成プログラム２１３の起動および実行制御を行うとともに、これらのプログラムによって生成された解析済み文書データ、構造インデックスおよび文字列インデックスをネットワーク１０５を介して文書検索サーバ１０２に転送する。

【０２６１】なお、本実施例ではフロッピーディスク２０５に格納された登録対象文書を入力として読み込む構成としたが、光磁気ディスク、追記型光ディスク等、他種の可搬型媒体から読み込む構成をとることもでき、ネットワーク１０５を介して転送されてくる文書を入力とする構成をとることもできる。さらに、本実施例では生成された解析済み文書データ、構造インデックスおよび文字列インデックスを文書検索サーバ１０２に転送するためにネットワーク１０５を使用するものとしたが、代わりにフロッピーディスク、光磁気ディスク、追記型光ディスク等の可搬型媒体を使用する構成をとることもできる。あるいは、文書登録サブシステム１０１と文書検索サーバ１０２を１台のコンピュータ上に実装し、データ転送を行わない構成をとることもできる。

【０２６２】図４５は、本発明の第四の実施例における文書登録処理の概略手順を示すＰＡＤ図である。本図に示す処理手順は、図３に示す第一の実施例における処理手順とほぼ同様であるが、ステップ３０４の直後にステップ４５０１が追加されている点が異なる。

【０２６３】ここで、ステップ４５０１では、ステップ３０４で生成した解析済み文書データを入力として文書構造正規化プログラム４４０１を実行する。文書構造正規化プログラム４４０１は、解析済み文書データの中から、検索対象としては不適切な構造および内容文字列を抽出してこれらを削除する。

【０２６４】図４６に、文書構造正規化プログラム４４０１の処理手順を示す。本プログラムは、起動されるとまずステップ４６０１において、正規化パラメタの指定の有無を確認する。正規化パラメタが指定されている場合にはステップ４６０２からステップ４６０８までに示す処理を実行し、正規化パラメタが指定されていない場合には何もせずに処理を終了する。

【０２６５】ここで、正規化パラメタとは、接続対象要素および削除対象要素の要素型名を指定するパラメタのことである。ここで言う接続対象要素とは、例えば文章の一部を強調表示するために用いられる非構造的な要素であって、検索時にはその要素の境界をまたいで文字列を検出する必要があるもののことである。また、削除対象要素とは、その内部に本来の文書内容とは異なる種類のデータを保持しており、検索時にはその内容を無視して文字列の検出を行うべきもののことである。削除対象要素の例としては、例えばテキスト中に参照先の文献へのリンクデータを埋め込むために用いられる要素などがある。

【０２６６】ステップ４６０２では、指定されている正規化パラメタを読み込む。ここで、正規化パラメタは、ユーザがキーボード２０２から入力するか、または予め特定のファイルに書き込んでおくなどして指定する。接続対象要素および削除対象要素の要素型名は、それぞれ複数個指定してもよく、あるいは省略してもよい。次に、ステップ４６０３では、解析済み文書データを解析済み文書データ格納領域２１７から読み込む。

【０２６７】次に、ステップ４６０４では、解析済み文書データの木構造を順次たどりつつ、すべての要素ノードについて、ステップ４６０５からステップ４６０７までに示す処理を繰り返す。そして、すべての要素ノードについて該処理を終了した後、ステップ４６０８に進む。

【０２６８】ステップ４６０５では、現在着目している要素ノードについて、正規化パラメタが何を指定しているかを判定する。該着目要素の要素型名が接続対象として指定されている場合にはステップ４６０６に進み、該要素ノードを削除するとともに、該要素の内部に含まれるすべての文字列データを、該要素の前後にある文字列データに接続する。また、該着目要素の要素型名が削除対象として指定されている場合にはステップ４６０７に進み、該要素ノードおよびその下位にあるすべてのノードを削除する。

【０２６９】前記のごとくステップ４６０５以下の処理を行った後、ステップ４６０８に進み、正規化対象要素群を処理して更新された解析済み文書データを、解析済み文書データ格納領域２１７に再び格納する。

【０２７０】図４７は、正規化処理の具体的な例を示す図である。

【０２７１】図４７において、４７０１は正規化パラメタの一例を示しており、ここでは接続対象要素の要素型名として“bold”と“italic”の二つ、削除対象要素の要素型名として“link”と“index”の二つが指定されている。この場合、解析済み文書データ中に４７０２に示すような構造を持つ部分があると、要素“bold”に対して前記ステップ４６０６に示す接続処理が行われ、そ

の結果は４７０３に示す構造となる。また、解析済み文書データ中に４７０４に示すような構造を持つ部分があると、要素“link”に対して前記ステップ４６０７に示す削除処理が行われ、その結果は４７０５に示す構造となる。

【０２７２】以上説明したように、本実施例では、解析済み文書データに対して前記正規化処理を行った後に構造インデックスへの登録以下の処理が行われるので、登録対象文書中に含まれていた非構造的な要素に妨げられることなくテキストの検索を行うことができる。

【０２７３】なお、本実施例における正規化処理を採用した場合においても、前記第二の実施例同様に正順・逆順二つの構造インデックスを用意することにより、出現順を後ろから辿った構造指定も可能とすることができる。

【０２７４】以上が、本発明の第四の実施例の説明である。

【０２７５】（第五の実施例）本発明を利用した第五の実施例について、図面を用いて説明する。

【０２７６】図４８は、本実施例における文書登録サブシステム１０１の構成を示す図である。

【０２７７】本システム構成は、そのハードウエア構成においては、図２に示す第一の実施例の文書登録サブシステム１０１と変わらない。ただし、主メモリ２０７に格納される文書登録プログラムの１つである、構造インデクス作成プログラム２１１を、メタ構造インデクス作成プログラム４８０１に変更している。さらに、磁気ディスク装置２０８には、構造インデクス格納領域２１８の代わりにメタ構造インデクス格納領域４８０２を作成し、種別定義テーブル格納領域４８０３を追加する。

【０２７８】メタ構造インデクス作成プログラム４８０１は、文書構造解析プログラム２１０の出力結果である解析済み文書データ２１７を入力として、様々な構造を持つ登録文書の全ての文書構造を一括管理するメタ構造インデクスを出力する。

【０２７９】図４９にメタ構造インデクスの例を示す。本図は、"論文"という種別の最上位構造を持つ構造インデクス１（４９０１）と"報告書"という種別の最上位構造を持つ構造インデクス２（４９０２）をルートメタノード４９０３によって１つの木構造にまとめたものである。この木構造をメタ構造インデクス４９０４と呼ぶ。

【０２８０】つまり、構造インデクスは、最上位構造の種別が一致する登録文書の構造との重ね合わせにより作成していくため、最上位構造の種別が同じ登録文書群毎に作成することになる。メタ構造インデクスとは、これらの異なる最上位構造を持つ全ての構造インデクスの最上位構造を、１つのルートメタノードに接続することで、１つのインデクスにまとめたものである。

【０２８１】ルートメタノードとは、複数の構造インデクスの最上位構造をまとめる仮の上位構造である。つまり、ルートメタノードから、複数の構造インデクスを辿るために存在しており、ルートメタノードに対応する構造が登録文書に存在するわけではない。

【０２８２】ルートメタノードは、構造インデクスの要素メタノードと同じく、下位要素である複数の構造インデクスの最上位構造の数および各構造インデクスへのリンクの情報を持つ。

【０２８３】上記のメタ構造インデクスを格納するのが、メタ構造インデクス格納領域４８０２である。

【０２８４】次に、種別定義テーブル４８０３について説明する。

【０２８５】種別定義テーブルは、構造化文書の各構造に付けられた要素型名とその意味を表わす種別との対応を定義したテーブルである。ユーザがキーボード２０２からテキストエディタを利用して作成するなどの方法で、あらかじめ磁気ディスク装置２０８に格納しておく。

【０２８６】図５０は種別定義テーブル４８０３の内容を示す図である。種別とは、例えば"論文"、"PAPER"のように、名称が違っていても同じ意味を持つ要素型名を代表する名称である。種別定義テーブルは、複数の要素型名に付けられた種別を管理しており、本テーブルにおいて同じ種別が定義されている要素型名を持つ構造は、同じ種別であると判断する。

【０２８７】本図に示したように、種別定義テーブルには、種別５００１と要素型名数５００２と要素型名５００３の３つの情報を格納する。種別５００１が、複数の要素型名に共通に付けられる種別である。要素型名数５００２は、各種別に対応する要素型名の数である。さらに、要素型名５００３には、要素型名数に記載された数の要素型名を列挙する。

【０２８８】本テーブルを参照することにより、要素型名から種別の情報を得ることができる。また、逆に、種別から要素型名を得ることができる。さらに、本テーブルに記載されない要素型名は、その要素型名をそのまま種別とする。

【０２８９】本実施例において、種別定義テーブル４８０３に記載される種別と要素型名は、１対多の対応を満たす。つまり、"書誌"という要素型名は一意に１つの種別を持つ。これは、種別定義テーブルをメタ構造インデクスごとに作成しており、メタ構造インデクスの各メタノードは、種別および出現位置により区別されるためである。要素型名によって一意に種別が決まらなければ、構造インデクスの作成において、要素型名から得られる複数の種別のうちいずれを用いるかが決まらないためである。

【０２９０】図５１は、本実施例における文書登録処理の概略手順を示すＰＡＤ図である。本図は、図３に示す第一の実施例の登録処理の概略手順とほぼ同じであるが、図３におけるステップ３０５の代わりにステップ５

51

１０１を実行する点と、ステップ３０８の代わりにステップ５１０２を実行する点が異なる。

【０２９１】ステップ５１０１では、メタ構造インデクス作成プログラム４８０１を呼び出す。メタ構造インデクス作成プログラム４８０１では、登録済みのメタ構造インデクスをメタ構造インデクス格納領域４８０２から読み出し、ステップ３０４で得られた解析済み文書データの持つ構造情報を、このメタ構造インデクスに登録し、更新されたメタ構造インデクスをメタ構造インデクス格納領域４８０２に格納する。

【０２９２】ステップ５１０２では、登録文書全ての解析済み文書データ２１７および更新されたメタ構造インデクス４８０４、文字列インデクス２２０を文書検索サーバ１０２に転送する。

【０２９３】図５２は、ステップ５１０１でメタ構造インデクスを作成する処理の詳細内容を示すＰＡＤ図である。本処理は、図９に示した第一の実施例における構造インデクス作成プログラムの処理とほぼ同じ内容であるが、登録先をメタ構造インデクスに変更していることから、以下の点で図９と異なる。

【０２９４】まず、ステップ９０４の解析済み文書データ読み込み処理を実行する。

【０２９５】次に、ステップ９０１の代わりにステップ５２０１を実行する。ステップ９０１は、構造インデクスそのものの有無を判定しているが、ステップ５２０１では、構造インデクスは１つのメタ構造インデクスの一部であり、最上位構造の種別毎に作成することから、メタ構造インデクス中に、登録文書の最上位構造と一致する構造インデクスが存在するか否かをチェックする処理に変更する。

【０２９６】ステップ９０４をステップ５２０１の前に実行するのは、本処理において、登録文書の最上位構造の情報が必要なためである。ここで、最上位構造の種別が一致する構造インデクスが存在しない場合は、新しく初期構造インデクスを生成するステップ９０２の処理を実行し、存在する場合は、該構造インデクスを読み込むステップ９０３の処理を実行する。また、本ステップにおいて、種別の比較をする前に、種別定義テーブル４８０３を参照し、登録文書の最上位構造の要素型名を種別に変換してから比較を行う。

【０２９７】さらに、ステップ９０６の代わりにステップ５２０２を実行する。ステップ５２０２では、種別定義テーブル４８０３を参照して、解析済み文書データの識別子名を種別に変換してから、図１１を用いて前述したステップ９０６の処理内容と同じ手順で、構造インデクス中の対応するメタノードの有無のチェックを行う。

【０２９８】さらに、ステップ９０８の代わりにステップ５２０３を実行する。ステップ５２０３では、文脈識別子を割り当てる際に、メタ構造インデクス全体で一意にメタノードを識別できる文脈識別子を与える。したがって、構造化全文データ作成プログラム２１２において、各構造のテキスト列に与えられる文脈識別子は、メタ構造インデクス中のメタノードを一意に決定するものである。ステップ９０８の処理に加えて、構造インデクスの識別情報を文脈識別子に加えるなどの方法により、実現可能である。

52

【０２９９】さらに、ステップ９０５による繰り返し処理の後に、ステップ５２０４を実行する。

【０３００】ステップ５２０４では、ステップ９０２で構造インデクスを新規に作成した場合に、メタ構造インデクスのルートメタノードに新規に作成した構造インデクスの最上位のメタノードを接続し、メタ構造インデクスに新規に作成した構造インデクスを組み込む処理を行う。

【０３０１】さらに、ステップ９１１の代わりにステップ５２０５を実行する。ステップ５２０５では、作成したメタ構造インデクスをメタ構造インデクス格納領域４８０２に格納する。

【０３０２】その他のステップの処理は、図９を用いて前述した内容と同じである。

【０３０３】図５３、図５４は、上記のステップ５１０１で作成するメタ構造インデクスの例である。図５３は、最上位構造の種別が一致する構造インデクスが存在する場合の例を示している。また、図５４は、最上位構造の種別が一致する構造インデクスが存在しない場合の例を示している。

【０３０４】図５３では、まずメタ構造インデクス５３０１に存在する構造インデクス５３０２と登録文書の構造解析結果の木構造５３０３との比較を行う。この例では登録文書の最上位構造である"文書"と一致する種別の最上位構造を持つ構造インデクス５３０２が存在するため、この構造インデクス５３０２に対して、登録文書の木構造５３０３の重ね合わせを行う。この場合、"日付"のノード５３０４が構造インデクス５３０２に存在しないため、構造インデクス５３０２に、"日付"のノードを追加して、更新された構造インデクス５３０５を作成する。構造インデクスの更新に伴い、メタ構造インデクス５３０１も更新される（５３０６）。

【０３０５】図５４では、まずメタ構造インデクス５４０１に存在する構造インデクス５４０２と登録文書の構造解析結果の木構造５４０３との比較を行う。この例ではメタ構造インデクス中の構造インデクスの最上位構造は、"論文"（５４０４）しかなく、登録文書の最上位構造である"報告書"（５４０５）と一致する最上位構造を持つ構造インデクスは存在しない。このため、登録文書の木構造と同じ構造を持つ、構造インデクス５４０６を新規に作成する。さらに、作成した構造インデクス５４０６を、ルートメタノード５４０７に接続することで、メタ構造インデクスに構造インデクス５４０６を追加する。構造インデクスの追加に伴い、メタ構造インデクス

５４０１も更新される（５４０８）。

【０３０６】上記のように、最上位構造が一致する構造インデクスが存在する場合は、この構造インデクスに対する重ね合わせを行い、存在しない場合は、新たに構造インデクスを作成した上で、ルートメタノードに接続することで、メタ構造インデクスを更新する。

【０３０７】本実施例の文書登録サブシステム１０１が、第一の実施例の文書登録サブシステムと異なる点は以上であり、その他の構成および処理内容は全て同じである。

【０３０８】次に、本発明の第五の実施例における文書検索サーバ、すなわち図１の１０２について説明する。

【０３０９】図５５は、本実施例における文書検索サーバ１０２の構成を示す図である。

【０３１０】本システム構成は、そのハードウエア構成においては、図１７に示す第一の実施例の文書検索サーバ１０２の構成と変わらない。

【０３１１】ただし、主メモリ２０７に格納される文書検索処理プログラムのうち検索条件解析プログラム１７０１をメタ構造インデクス対応検索条件解析プログラム５５０１に変更し、さらに構造インデクス格納領域２１８の代わりにメタ構造インデクス格納領域４８０２を作成し、磁気ディスク装置２０８に種別定義テーブル格納領域４８０３を追加する点が異なる。

【０３１２】メタ構造インデクス対応検索条件解析プログラム５５０１は、検索クライアント１０３および１０４から受信した検索リクエスト中に含まれる検索条件式を解析し、文字列インデクス検索プログラム１７０２によって直接検索可能な条件指定に翻訳する。ここで、検索条件解析プログラム１７０１が、構造インデクスを利用して、検索条件式を解析するのと違い、メタ構造インデクス対応検索条件解析プログラム５５０１では、メタ構造インデクスおよび種別定義テーブルを利用して検索条件式を解析する。

【０３１３】また、メタ構造インデクス格納領域４８０２には、本実施例において前述した、文書登録サブシステム１０１で作成したメタ構造インデクスを格納する。種別定義テーブル４８０３は、ユーザによって文書登録サブシステム１０１に登録された種別定義テーブルと同じ内容である。

【０３１４】図５６は、本実施例における検索サーバの処理の概略手順を示すＰＡＤ図である。本図は、図１８に示す第一の実施例の検索サーバの処理の概略手順とほぼ同じである。ただし、図１８におけるステップ１８０５の代わりにステップ５６０１を実行する点と、ステップ１８０６の代わりにステップ５６０２を実行する点が異なる。

【０３１５】ステップ５６０１では、文書登録サブシステム１０１から、メタ構造インデクスおよび文字列インデクスを受信し、これらをそれぞれメタ構造インデクス格納領域４８０２および文字列インデクス格納領域２２０に格納する。これらのメタ構造インデクスおよび文字列インデクスは、ステップ１８０４で追加登録した、新規に登録された文書群の内容を反映して更新したものである。

【０３１６】ステップ５６０２では、メタ構造インデクス対応検索条件解析プログラム５５０１を実行し、文書検索リクエスト中で指定されている検索条件を解析し、該検索条件を、文字列インデックス検索プログラム１７０２によって直接処理可能な条件指定（以下、これを展開済み検索条件データと呼ぶ）に変換する。

【０３１７】その他のステップの処理は、第一の実施例において、図１８を用いて前述した内容と同じである。

【０３１８】図５７は、図５６におけるステップ５６０２の詳細、すなわち本実施例におけるメタ構造インデクス対応検索条件解析プログラム５５０１の処理手順を示すＰＡＤ図である。本図は、図１９に示す第一の実施例の検索条件解析プログラム１７０１の処理手順を示すＰＡＤ図とほぼ同じである。図１９におけるステップ１９０２の代わりにステップ５７０１を実行する点と、ステップ１９０３の代わりにステップ５７０２を実行する点が異なる。

【０３１９】ステップ５７０１では、メタ構造インデックス格納領域４８０２からメタ構造インデックスを読み込む。

【０３２０】次に、ステップ５７０２では、メタ構造インデックスを参照して、前記構造条件を満たす構造内に含まれるすべての文字列データの文脈識別子集合を求める。ここで検索条件で指定された構造条件が、構造の種別により指定される場合は、そのまま、メタ構造インデクスを辿ることで、構造条件を満たす構造内に含まれる文字列データの文脈識別子を得ることができる。要素型名で指定される場合は、種別定義テーブル４８０３を参照して種別に変換してから、メタ構造インデクスを辿って、構造条件を満たす構造内に含まれる文字列データの文脈識別子を得る。

【０３２１】メタ構造インデクスの最上位構造は、各文書の最上位構造を接続したルートメタノードであるため、対応する構造条件は存在しない。

【０３２２】その他のステップは、第一の実施例において、図１９を用いて前述した内容と同じである。

【０３２３】図５８は、メタ構造インデクス対応検索条件解析プログラム５５０１の処理過程における、展開済み解析条件データの生成例を示す図である。

【０３２４】図５８において、５８０１は、文書検索リクエスト中で指定された検索条件の一例である。検索条件５８０１は、構造条件指定“論文/書誌/タイトル”と、文字列条件指定“ガード”とから構成されている。前記検索条件は、“論文”要素の直接の下位にある“書誌”要素の直接の下位にある“タイトル”要素内に、文

字列“ガード”が出現するケースを検索すべきことを指定している。

【０３２５】ここでは、構造条件に種別を指定した場合について説明する。要素型名を指定した場合は、種別定義テーブル４８０３を参照して、種別に変換した後に以下の処理を行なう。

【０３２６】さらに、構造条件に要素型名、種別を混在して指定するために、種別の場合は構造条件の前に"Type:"などの識別情報を付加する。このため、検索クライアント１０３、１０４において、第一の実施例で図２７を用いて説明したステップ２７０２およびステップ２７０３で、ユーザ指示および文書検索リクエストに上記の識別情報を追加する。ただし、本設定を行なっても、メタ構造インデクスには種別の情報しか持たないため、要素型名による構造条件でメタ構造インデクスから構造条件を満たす構造内に含まれる文字列データの文脈識別子が得られるわけではない。

【０３２７】ここで、メタ構造インデックスの内容が５８０２に示すとおりであったとすると、前記ステップ５７０２において、この構造インデックスを参照することにより、前記構造条件指定を満たす“タイトル”要素の文脈識別子はＥ３であることがわかる。従って、この段落の下位にある文字列データ、すなわち文脈識別子がＣ１である文字列データ内に、文字列“ガード”が出現するケースを検索すればよいことがわかる。ただし、検索に用いる文字列インデックスには、長さ２の部分文字列についてのみ、その出現位置が登録されているので、３文字からなる前記指定文字列を直接検索することはできない。そこで、ステップ１９０５において、前記指定文字列を分解して長さ２の部分文字列からなるリストを生成する。前記のとおり指定文字列が“ガード”だった場合、抽出される部分文字列は“ガー”および“ード”となる。

【０３２８】この結果、前記ステップ１９０７において、５８０３に示す展開済み検索条件データ、すなわち文脈識別子集合が｜Ｃ１｜、指定文字列が“ガード”、部分文字列リストが｜“ガー”，“ード”｜であるデータが生成される。

【０３２９】以上説明したように、本実施例に示した構成によると、複数の構造を持つ文書の検索を一括して行なうことができる。また、検索条件において、種別および要素型名を指定した構造条件が可能となる。

【０３３０】以上が、本発明を利用した第五の実施例の説明である。

【０３３１】（第六の実施例）次に、本発明を利用した第六の実施例を図面を利用して説明する。

【０３３２】第六の実施例において第五の実施例と異なる点は、文書登録の処理では、種別定義テーブルを利用せず、要素型名をそのまま構造の種別としてメタ構造インデクスを作成し、検索処理の際に、種別を用いた構造

条件を含む構造指定検索条件を、要素型名を指定した構造条件に変換した上で、検索処理を実施する点である。これにより、種別による構造条件と要素型名による構造条件のいずれも指定可能である。

【０３３３】本実施例における文書登録サブシステム１０１のシステム構成は、図４８に示す第五の実施例の文書登録サブシステム１０１のシステム構成と同じである。

【０３３４】ただし、第五の実施例で図４８に示したメタ構造インデクス作成プログラム４８０１のうち、図５２を用いて説明した、ステップ５２０１とステップ５２０２の処理内容を一部変更する。すなわち、ステップ５２０１とステップ５２０２において、種別定義テーブル４８０３を参照して要素型名から種別への変換を行う処理は実行せず、要素型名をそのまま種別とみなして、構造インデクスへの登録処理を行う。ただし、種別定義テーブル４８０３を作成し、検索サーバへ転送する処理は変わらない。

【０３３５】以上が、第六の実施例における文書登録サブシステム１０１の、第五の実施例における文書登録サブシステムからの変更点である。その他の構成、処理内容は全て同じである。

【０３３６】本実施例における文書検索サーバ１０２のシステム構成も図５５と同じである。

【０３３７】ただし、第五の実施例で図５５に示したメタ構造インデクス作成プログラム５５０１のうち、図５７を用いて説明したステップ５７０２の処理が一部変更される。

【０３３８】つまり、ステップ５７０２において、検索条件で指定された構造条件が、構造の種別により指定されていれば、種別定義テーブル４８０３を参照して、該種別に対応する全ての要素型名を取得し、得られた要素型名のＯＲにより作成した構造条件に変更する。メタ構造インデクスを辿ることで、作成した構造条件を満たす構造内に含まれる文字列データの文脈識別子を得ることができる。要素型名で指定された場合は、そのまま、メタ構造インデクスを辿って、構造条件を満たす構造内に含まれる文字列データの文脈識別子を得る。

【０３３９】図５９に、本実施例のステップ５７０２において、種別による構造条件が指定された場合に要素型名に変更して生成する構造条件を示す。本図に示すように、構造条件を構成する種別ごとに、要素型名を取得し、構造条件の各階層を１つあるいは複数の要素型名の論理和（以下ＯＲ）で記述する構造条件を生成する。複数の要素型名のＯＲは、構造条件に「｜種別または要素型名，種別または要素型名，…｜」のように"｜　｜"内にで複数の種別または要素型名を列挙するなどの方法により指定する。

【０３４０】ユーザが検索条件を種別を指定する際には、種別名の前に"Type:"などの識別情報を記載するこ

57

とで、その名称を種別として、要素型名に変更する。何も指定せずに検索条件を記述する場合は、要素型名であると判断し、そのまま、構造条件を利用して、構造インデクスから適合する構造の文脈識別子を取得する。あるいは、同じ名称の要素型名が存在しない種別であれば、曖昧性がないため、"Type:"などの識別指定を省略してもよい。

【０３４１】図５９では、「Type:属性/Type:題目」という構造条件（５９０１）を、構造条件の変換処理（５９０２）により、種別定義テーブル４８０３を参照して、要素型名を用いた構造条件（５９０３）に変換する。

【０３４２】検索クライアント１０３、１０４の変更点は、第五の実施例で図５８を用いて説明した通りである。ただし、本実施例では、要素型名を指定した構造条件は、要素型名が一致する構造内の文字列データの文脈識別子が取得でき、種別を指定した構造条件は、種別が一致する構造内の文字列データの文脈識別子が取得できる。

【０３４３】以上の処理により、本実施例では、検索条件に種別による構造条件と要素型名による構造条件を組み込むことが可能となる。

【０３４４】さらに、本実施例の方法の第五の実施例の方法と比較したメリットは、構造インデクスが、要素型名毎にメタノードが作成されるため、種別の変更が任意に可能であるということである。例えば、クライアント毎に種別定義テーブルを作成して検索サーバに転送した後に、種別定義テーブルを指定した検索条件を設定するなどの処理方法も可能であり、柔軟な種別の設定が実現できる。第五の実施例では、種別定義テーブルの変更に対応するためには、変更時点までに作成した、メタ構造インデクス、文字列インデクスの再作成が必要となる。

【０３４５】また、本実施例の方法のデメリットとしては、第五の実施例に比べて、生成されるメタ構造インデクスが大きくなることがある。メタ構造インデクスにおいて、種別ごとにメタノードを作成する方が、要素型名ごとに作成する場合に比べて、メタノード数を削減できるためである。

【０３４６】以上が、本発明を利用した第六の実施例の説明である。

【０３４７】（第七の実施例）次に第七の実施例として、異なる文書構造を持つ文書群を、メタ構造インデクスを使わずに１つの構造インデクスを利用して構造指定検索するためのシステム構成および処理手順を説明する。

【０３４８】図６０は、本実施例における文書登録サブシステム１０１のシステム構成である。本システム構成は、そのハードウエア構成においては、図２に示す第一の実施例の文書登録サブシステム１０１と変わらない。ただし、主メモリ２０７に格納する文書登録プログラム

58

にルートノード付加プログラム６００１を追加する。

【０３４９】ルートノード付加プログラム６００１の処理内容を図６１を用いて説明する。ルートノード付加プログラム６００１は、文書構造解析プログラム２１０の出力結果である解析済み文書データ６１０１を解析済み文書データ格納領域２１７から読み出し、該解析済み文書データの最上位ノードの上位ノードとして、固定の種別を持つノードを付加した、ルートノード付加解析済み文書データ６１０２を作成し、解析済み文書データ格納領域２１７に格納する。これにより、読み出した解析済み文書データ６１０１をルートノード付加解析済み文書データ６１０２に置き換える。

【０３５０】本実施例の文書登録サブシステム１０１において、上記の処理以外は、第一の実施例の文書登録サブシステムの構成および処理と全て同じである。

【０３５１】図６２は、本実施例における文書検索サーバ１０２のシステム構成である。本システム構成は、そのハードウエア構成においては、図１７に示す第一の実施例の文書検索サーバ１０２と変わらない。ただし、主メモリ２０７に格納する文書登録プログラムに検索条件修正プログラム６２０１を追加する。

【０３５２】検索条件修正プログラム６２０１は、構造条件が最上位構造から指定された場合は、文書登録時に登録文書の解析済み文書データの最上位構造に追加したrootを構造条件に追加する処理が加わる。その他の場合は、検索条件を変更することはない。

【０３５３】図６３は、検索条件修正プログラム６２０１の処理内容を示すＰＡＤ図である。

【０３５４】まず、ステップ６３０１で、検索条件中の構造条件の有無をチェックする。構造条件が存在するなら、ステップ６３０２に進み、存在しないなら、検索条件を何も変更せず、検索条件修正プログラム６２０１を終了する。

【０３５５】ステップ６３０２で、構造条件が最上位構造から指定されているか否かをチェックする。最上位構造から指定されている場合は、ステップ６３０３に進む。最上位構造から指定されていない場合は、検索条件を何も変更せず、検索条件修正プログラム６２０１を終了する。

【０３５６】ステップ６３０３で、構造条件を変更し、最上位構造のrootを指定した検索条件とする。

【０３５７】ステップ６３０４で、変更した検索条件を出力する。１７０２以降の処理内容は、図１７を用いて前述した第一の実施例の検索サーバ１０２の処理内容と同じである。

【０３５８】図６４は、検索条件修正の結果である。本図に示すように構造条件の構造条件に、最上位構造が指定される場合は、rootという構造を追加した構造条件を生成する。

【０３５９】上記の処理により検索条件を変更する処理

以外は、全て第一の実施例の検索サーバ１０２の構成および処理内容と同じである。

【０３６０】次に、第七の実施例における検索クライアント１０３、１０４の処理内容について説明する。

【０３６１】本実施例において、検索クライアント１０３、１０４のシステム構成は、図２５の第一の実施例の検索クライアントのシステム構成と同じである。

【０３６２】ただし、検索結果表示プログラム２５０２の処理手順を示す図２８のＰＡＤ図のステップ２８１５において、解析済み文書データを書式化して表示する際に、文書登録サブシステム１０１において、追加されたルートノードは削除してから書式化して表示する。つまり、登録文書を構造解析した結果である解析済み文書データに変換してから表示するように変更する。これにより、ユーザからは、解析済み文書データに追加されたルートノードの存在は見えないことになる。

【０３６３】以上が第七の実施例における、第一の実施例の処理からの変更点である。その他の構成および処理内容は、第一の実施例の構成および処理内容と同じである。

【０３６４】以上の処理により、解析済み文書データが登録文書と異なる点を除いて、構造インデクスを利用してメタ構造インデクスを利用した場合と同様に様々な文書構造を持つ文書群に対する一括した構造指定検索が可能となる。

【０３６５】（第八の実施例）次に、複数の構造において同じ種別を持つ構造をまとめて効率的に検索するための別名構造インデクスの作成方法およびこれを用いた検索処理について説明する。

【０３６６】図６５は、別名構造インデクス６５０１の構成および別名構造インデクス６５０１とメタ構造インデクス６５０２との関係および、別名構造インデクスを作成するために用いる別名定義テーブル６５０３の内容を示した図である。

【０３６７】別名構造インデクスは、必ずしも構造インデクスのように文書全体の構造を辿るためのインデクスを作成するわけではなく、文書構造の部分構造を構造インデクスから切り出し、切り出した部分構造を重ね合わせて作成する。

【０３６８】図６５に示すように、異なる文書構造の書誌に関する情報を切り出し、メタ構造インデクスを構成するメタノードの文脈識別子を管理することで、検索条件として個々の構造を指定しなくても、１つの別名を指定した構造条件を設定することで、その別名に対応する全てのメタ構造インデクス中のメタノードの文脈識別子を得ることができる。

【０３６９】別名定義テーブル６５０３には、本図で示したように、別名６５０４と構造定義数６５０５と構造定義６５０６が格納される。

【０３７０】別名６５０４は、別名構造インデクスを参照する際の名称が格納される。構造定義数６５０５には、別名登録された構造定義数を記述する、構造定義６５０６には、別名６５０４で代表される検索条件中の構造条件を構造定義数分列挙する。

【０３７１】別名構造インデクスは、いくつかの構造定義によって指定される構造インデクス中のメタノードの文脈識別子をあらかじめ取得しておき、構造条件から検索条件を満たす構造に含まれる文字列データの文脈識別子を高速に取得するためのインデクスである。

【０３７２】別名構造インデクスの各ノードには、構造インデクスのメタノードと同じように、論理構造をあらわすためのリンク情報とメタノードの文脈識別子を持つ。ただし、メタノードの文脈識別子には、別名として定義された構造に含まれる文字列データのメタノードの文脈識別子をすべて格納する。

【０３７３】図６６は、本実施例における文書登録サブシステム１０１のシステム構成を示す図である。

【０３７４】本実施例における文書登録サブシステム１０１のシステム構成は、図４８に示す第五の実施例の文書登録サブシステム１０１のシステム構成とそのハードウェア構成に関しては変わるところはない。

【０３７５】ただし、主メモリ２０７に格納する文書登録プログラムに別名構造インデクス作成プログラム６６０１を追加し、磁気ディスク２０８には、別名構造インデクス格納領域６６０２および別名定義テーブル６６０３を追加する点が異なる。

【０３７６】別名構造インデクス作成プログラム６６０１は、別名定義テーブル格納領域６６０３から別名定義テーブルを読み出す。さらに、メタ構造インデクス作成プログラム４８０１で作成するメタ構造インデクスをメタ構造インデクス格納領域４８０２から読み出す。読み込んだ情報を元に、別名構造インデクスを作成し、別名構造インデクス格納領域６６０２に格納する。

【０３７７】図６７は、本実施例における文書登録サブシステム１０１の概略処理手順を示すＰＡＤ図である。本実施例の処理手順は、図５１を用いて前述した、第五の実施例の文書登録サブシステム１０１の概略処理手順とほぼ同じである。ただし、ステップ５１０１のあとで、ステップ６７０１を実行する点と、ステップ５１０２の代わりにステップ６７０２を実行する点が異なる。

【０３７８】ステップ６７０１では、別名構造インデクス作成プログラム６６０１を実行し、文書の登録により更新されたメタ構造インデクスの情報を参照し、別名構造インデクスの内容を更新する。

【０３７９】ステップ６７０２では、全ての解析済み文書データ、メタ構造インデクス、別名構造インデクス、文字列インデクスを文書検索サーバ１０２に転送する。

【０３８０】図６８は、図６７のステップ６７０１の詳細手順を示すＰＡＤ図である。本図を用いて、別名構造インデクスの作成手順を説明する。

61

【０３８１】まず、ステップ６８０１で、別名として作成する構造を定義した別名定義テーブル６６０３を読み出す。別名定義テーブル６６０３は、ユーザがキーボード２０２からテキストエディタなどを利用して作成する。あるいは、構造インデクスから、異なる階層に存在する同じ種別の構造を抽出して、この情報を元に別名定義テーブルを作成するプログラムによって作成する。

【０３８２】次に、ステップ６８０２で、上記ステップ６８０１で読み出した別名定義テーブル６６０３を用いて、構造インデクス中から、該構造情報に適合するメタノードを抽出する。これは、図５７の５７０２ステップで前述した、第五の実施例における文書検索処理における検索条件に適合するメタ構造インデクスの取得処理と同じ処理で実現できる。

【０３８３】ステップ６８０３では、得られたメタノードの文脈識別子を管理するテーブルを作成し、これを別名構造インデクスに登録する。

【０３８４】ステップ６８０４では、階層構造がある別名について、階層構造を表現するためにノード同士を接続する。別名構造インデクスに登録する別名は、"書誌/題目"のように階層的な別名を指定することが可能である。この場合、まず、構造インデクスから、"書誌"の種別情報を持つメタノードを抽出した上で、その子ノードから"題目"の種別情報を持つメタノードを抽出し、このメタノードの文脈識別子を管理する文脈識別子管理テーブルを作成して、別名構造インデクスに登録する。さらに、この過程で得られる"書誌"の種別情報を持つメタノードについても文脈識別子管理テーブルを作成し、別名構造インデクスの"書誌"に格納することで、階層的な構造を持った、別名構造インデクスを作成する。

【０３８５】図６９に、本実施例における全文検索サーバ１０２のシステム構成図を示す。

【０３８６】本構成図は、図５５を用いて前述した第五の実施例における全文検索サーバ１０２のシステム構成図とそのハードウェア構成においては同じである。ただし、主メモリ２０７にメタ構造インデクス対応検索条件解析プログラム５５０１の代わりに別名構造インデクス対応検索条件解析プログラム６９０１を格納する点と、磁気ディスク２０８に別名構造インデクス格納領域を追加する点が異なる。

【０３８７】図７０は、本実施例における検索処理の概要を示すＰＡＤ図である。

【０３８８】本図に示した処理は、図５６に示した第五の実施例の検索処理とほぼ同じであるが、ステップ５６０１の代わりにステップ７００１を実行する点とステップ５６０２の代わりにステップ７００２を実行する点が異なる。

【０３８９】ステップ７００１では、メタ構造インデクス、別名構造インデクス、文字列インデクスを文書登録サブシステム１０１から受信して、それぞれメタ構造インデクス格納領域４８０２、別名構造インデクス格納領域６６０２、文字列インデクス格納領域２２０に格納する。

62

【０３９０】ステップ７００２では、別名構造インデクス対応検索条件解析プログラム６９０１を実行する。

【０３９１】図７１は、ステップ７００２の処理の詳細すなわち、別名構造インデクス対応検索条件解析プログラム６９０１の処理手順を示すＰＡＤ図である。

【０３９２】本図の処理は、図５７に示す第五の実施例のメタ構造インデクス対応検索条件解析プログラムの処理手順とほぼ同じである。

【０３９３】ただし、構造条件の有無を判定するステップ１９０１の代わりに、構造条件の有無および別名指定か否かを判定するステップ７１０１を実行する点と、ステップ７１０１で別名と判定された場合は、ステップ７１０２、ステップ７１０３を実行する点が異なる。構造条件が種別もしくは要素型名の場合は、第五の実施例と同じく、ステップ５７０１とステップ５７０２を実行する。

【０３９４】ステップ７１０１では、構造指定検索の構造条件として、別名を利用しているか否かを判定する。構造条件で別名を利用する場合は、構造条件の先頭に、例えば"Alias:"という文字列を指定することで区別する。したがって、別名である"題目"を検索対象の構造として指定する場合は、"Alias:題目"と構造条件に記述しているか否かをチェックすることで判定することができる。

【０３９５】ステップ７１０２では、別名構造インデクスを読み込む。次にステップ７１０３で、別名構造インデクスを参照し、指定された構造条件を満たす文字列データの文脈識別子の集合を求める。別名インデクスに格納された、別名に対応するメタ構造インデクスのメタノードの下位にある文字列データのメタノードの文脈識別子を取得することで実現できる。

【０３９６】その他の処理は、図５７で示した、第五の実施例のメタ構造インデクス対応検索条件解析プログラムの処理手順と同じである。

【０３９７】本実施例における、検索サーバ１０２の構成および処理内容は、その他の点においては、全て第五の実施例における、全文検索サーバ１０２の処理と同じである。

【０３９８】以上が、本発明を利用した第八の実施例の説明である。

【０３９９】（第九の実施例）次に第九の実施例として、第五の実施例における種別定義テーブル４８０３の記載内容を変更し、各文書構造毎に要素型名の種別を指定できるようにする方法について説明する。

【０４００】図７２を用いて、本実施例における種別定義テーブル４８０３の格納情報について説明する。本図のように、DTD名称と要素型名を合わせてDTDおよび要素

型名の領域７２０１に格納することで、解析済み文書データの要素型名だけでなく、DTD名称との組み合わせで種別を決めることができる。これにより、例えば、「報告書」における「本文」の種別は「報告内容」であり、その他の文書の「本文」は、種別も「本文」のままとし、登録文書の文書構造に合わせた種別を定義することが可能となる。

【０４０１】本実施例における文書登録サブシステム１０１のシステム構成は、図４８に示す第五の実施例の文書登録サブシステムのシステム構成と同じである。さらに、本実施例における、文書登録サブシステム１０１の処理手順は、図５２のＰＡＤ図に示す第五の実施例における文書登録サブシステムの処理手順と同じである。ただし、ステップ５２０１における構造インデクスの最上位構造の取得の際に種別定義テーブル４８０３を参照して単に要素型名を種別に変換するのではなく、登録文書のＤＴＤと要素名の組み合わせに対応する種別を取得する点が異なる。さらに、ステップ５２０２において、構造インデクスの重ね合わせを行なう際に、種別定義テーブル４８０３を参照して、登録文書のＤＴＤと要素型名の組み合わせにより種別を得てから重ね合わせを行なう点が異なる。

【０４０２】その他の構成及び処理内容において、本実施例が第五の実施例と異なる点はない。

【０４０３】以上が、本発明を利用した第九の実施例である。

【０４０４】（第十の実施例）次に第十の実施例として、第五の実施例における種別定義テーブル４８０３を構造インデクス毎に管理し、メタ構造インデクス中の各構造インデクス毎に種別定義テーブルを参照して種別を得る方法について説明する。

【０４０５】本実施例における文書登録サブシステム１０１のシステム構成は、図４８に示す第五の実施例の文書登録サブシステムのシステム構成と同じである。ただし、本実施例では、種別定義テーブル４８０３を各構造インデクス毎に作成し、メタ構造インデクスには、各構造インデクスの最上位構造の種別に関する種別定義テーブルを持つ構成とする点が異なる。このような構成とすることで、１つの要素型名を構造インデクス毎に異なる種別に割り当てることができる。

【０４０６】これらの種別定義テーブルは、第五の実施例において、図５０に示す内容としても良いし、第九の実施例において、図７２で示す内容としても良い。以下の説明では、種別定義テーブルは図５０に示す第五の実施例の種別定義テーブルを用いる場合について説明するが、図７２で示した第九の実施例の種別定義テーブルを用いる場合でも同様の手順で実現可能である。

【０４０７】図７３は、本実施例におけるメタ構造インデクスと、種別定義テーブルの関連を示す図である。メタ構造インデクス７３０１のルートメタノード７３０２

に対応する、最上位構造種別定義テーブル７３０３を作成する。さらに、各構造インデクス毎に種別定義テーブルを作成する。本図では、論文の構造インデクス７３０４に対応する種別定義テーブル１（７３０５）を作成し、さらに報告書の構造インデクス７３０６に対応する種別定義テーブル２（７３０７）を作成する。このような構成にすることで、構造インデクス毎に種別を定義することができる。

【０４０８】本実施例における、文書登録サブシステム１０１の処理手順は、図５２に示す第五の実施例における文書登録サブシステムの処理手順を示したＰＡＤ図と同じである。ただし、ステップ５２０１における構造インデクスの最上位構造の取得の際には、最上位構造種別定義テーブル７３０３を参照して要素型名を種別に変換し、対応する構造インデクスを得る。さらにステップ５２０２において、ステップ５２０１で得られる構造インデクスに対応する種別定義テーブルを参照して、要素型名を種別に変換し構造インデクスの重ね合わせを行なう。例えば登録文書の種別が論文であれば、構造インデクス７３０４に対応する種別定義テーブル１（７３０５）を参照して、要素型名を種別に変換する。

【０４０９】その他の構成及び処理内容において、本実施例が第五の実施例と異なる点はない。

【０４１０】第九の実施例のように種別定義テーブルが図７２に示す構成であっても、第九の実施例で示した手順により要素型名だけでなくＤＴＤ名称との組み合わせで参照することで、同様に実現可能である。

【０４１１】以上が、本発明を利用した第十の実施例である。

【０４１２】

【発明の効果】以上説明したように、本発明による構造化文書の検索方法によれば、文書中に現れる論理要素の文書中における出現位置に関する条件を構造条件指定中に含めることができるので、複雑な論理構造を備えた多数の文書からなる文書データベース上においても精度の高い構造指定検索ができる。

【図面の簡単な説明】

【図１】本発明による文書検索システムの第一の実施例の全体構成を示す図である。

【図２】本発明の第一の実施例における文書登録サブシステムの構成を示す図である。

【図３】本発明の第一の実施例における文書登録処理の概略手順を示すＰＡＤ図である。

【図４】文書論理構造を定義するＤＴＤの一例を示す図である。

【図５】ＳＧＭＬによる構造化文書の記述例を示す図である。

【図６】ＳＧＭＬが表現する文書の論理構造を図形的に示した模式図である。

【図７】本発明の第一の実施例における文書構造解析プ

ログラムの処理手順を示すＰＡＤ図である。

【図８】文書構造テーブルのデータ構造を示す図である。

【図９】本発明の第一の実施例における構造インデックス作成プログラムの処理手順を示すＰＡＤ図である。

【図１０】本発明の第一の実施例における解析済み文書データの辿り順を示す図である。

【図１１】本発明の第一の実施例におけるノードとメタノードとの対応関係を示す図である。

【図１２】本発明の第一の実施例における構造インデックスの生成過程を示す図である。

【図１３】本発明の第一の実施例における構造化全文データ生成プログラムの処理手順を示す図である。

【図１４】本発明の第一の実施例における構造化全文データのファイル形式を示す図である。

【図１５】本発明の第一の実施例における文字列インデックス作成プログラムの処理手順を示すＰＡＤ図である。

【図１６】本発明の第一の実施例における文字列インデックスのデータ構造を示す図である。

【図１７】本発明の第一の実施例における文書検索サーバの構成を示す図である。

【図１８】本発明の第一の実施例における文書検索処理の概略手順を示すＰＡＤ図である。

【図１９】本発明の第一の実施例における検索条件解析プログラムの処理手順を示すＰＡＤ図である。

【図２０】本発明の第一の実施例における展開済み検索条件データの生成例を示す図である。

【図２１】本発明の第一の実施例における文字列インデックス検索プログラムの処理手順を示すＰＡＤ図である。

【図２２】本発明の第一の実施例における連接判定処理の実行例を示す図である。

【図２３】本発明の第一の実施例における検索結果データのデータ構造を示す図である。

【図２４】本発明の第一の実施例における検索結果データ転送処理の詳細手順を示すＰＡＤ図である。

【図２５】本発明の第一の実施例における文書検索クライアントの構成を示す図である。

【図２６】本発明の第一の実施例における検索クライアントの動作手順を示すＰＡＤ図である。

【図２７】本発明の第一の実施例における検索条件入力プログラムの処理手順を示すＰＡＤ図である。

【図２８】本発明の第一の実施例における検索結果表示プログラムの処理手順を示すＰＡＤ図である。

【図２９】本発明の第二の実施例における文書登録サブシステムの構成を示す図である。

【図３０】本発明の第二の実施例における文書登録処理の概略手順を示すＰＡＤ図である。

【図３１】本発明の第二の実施例における逆順構造インデックス作成プログラムの処理手順を示すＰＡＤ図である。

【図３２】本発明の第二の実施例における解析済み文書データの辿り順を示す図である。

【図３３】本発明の第二の実施例におけるノードとメタノードとの対応関係を示す図である。

【図３４】本発明の第二の実施例における逆順構造インデックスの生成過程を示す図である。

【図３５】本発明の第二の実施例における構造化全文データ生成プログラムの処理手順を示すＰＡＤ図である。

【図３６】本発明の第二の実施例における構造化全文データのファイル形式を示す図である。

【図３７】本発明の第二の実施例における文字列インデックスのデータ形式を示すＰＡＤ図である。

【図３８】本発明の第二の実施例における文書検索サーバの構成を示す図である。

【図３９】本発明の第二の実施例における文書検索処理の概略手順を示すＰＡＤ図である。

【図４０】本発明の第二の実施例における検索条件解析プログラムの処理手順を示すＰＡＤ図である。

【図４１】本発明の第二の実施例における展開済み検索条件データの生成例を示す図である。

【図４２】本発明の第二の実施例における文字列インデックス検索プログラムの処理手順を示すＰＡＤ図である。

【図４３】本発明の第三の実施例におけるノードとメタノードとの対応関係を示す図である。

【図４４】本発明の第三の実施例における文書登録サブシステムの構成を示す図である。

【図４５】本発明の第四の実施例における文書登録処理の概略手順を示すＰＡＤ図である。

【図４６】本発明の第四の実施例における文書構造正規化プログラムの処理手順を示すＰＡＤ図である。

【図４７】本発明の第四の実施例における正規化処理の具体例を示す図である。

【図４８】本発明の第五の実施例における文書登録サブシステムの構成を示す図である。

【図４９】本発明の第五の実施例におけるメタ構造インデクス作成の例を示す図である。

【図５０】本発明の第五の実施例における種別定義テーブルの内容を示す図である。

【図５１】本発明の第五の実施例における文書登録処理の概略手順を示すＰＡＤ図である。

【図５２】本発明の第五の実施例におけるメタ構造インデクス作成処理の概略手順を示すＰＡＤ図である。

【図５３】本発明の第五の実施例におけるメタ構造インデクスの更新処理の例１を示す図である。

【図５４】本発明の第五の実施例におけるメタ構造インデクスの更新処理の例２を示す図である。

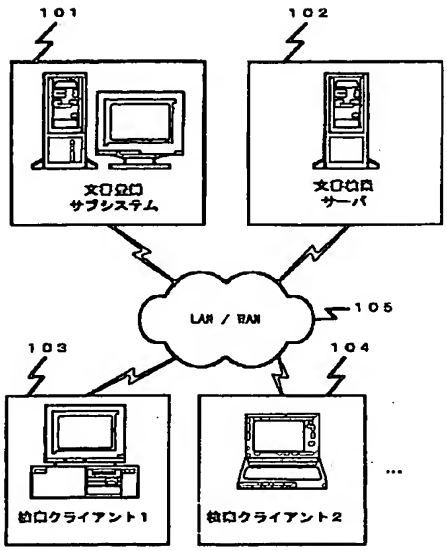
【図５５】本発明の第五の実施例における文書検索サー

バの構成を示す図である。
【図５６】本発明の第五の実施例における文書検索処理の概略手順を示すＰＡＤ図である。
【図５７】本発明の第五の実施例におけるメタ構造インデクス対応検索条件解析プログラムの処理手順を示すＰＡＤ図である。
【図５８】本発明の第五の実施例における展開済み検索条件データの生成例を示す図である。
【図５９】本発明の第六の実施例における構造条件変換の例を示す図である。
【図６０】本発明の第七の実施例における文書登録サブシステムの構成を示す図である。
【図６１】本発明の第七の実施例におけるルートノード付加プログラムの処理結果の例を示す図である。
【図６２】本発明の第七の実施例における文書検索サーバの構成を示す図である。
【図６３】本発明の第七の実施例におけるルートノード付加プログラムの処理手順を示す図である。
【図６４】本発明の第七の実施例における構造条件変換処理の内容を示す図である。
【図６５】本発明の第八の実施例における別名構造インデクスを示す図である。
【図６６】本発明の第八の実施例における文書登録サブシステムのシステム構成図である。
【図６７】本発明の第八の実施例における登録処理の概略手順のＰＡＤ図である。
【図６８】本発明の第八の実施例における別名構造インデクス作成処理手順を示すＰＡＤ図である。
【図６９】本発明の第八の実施例における文書検索サーバのシステム構成図である。
【図７０】本発明の第八の実施例における文書検索の概略処理を示すＰＡＤ図である。
【図７１】本発明の第八の実施例における別名構造インデクス対応検索条件解析プログラムの処理手順を示すＰＡＤ図である。
【図７２】本発明の第九の実施例における種別定義テーブルの内容を示す図である。
【図７３】本発明の第十の実施例におけるメタ構造インデクスと構造インデクス毎の種別定義管理テーブルの対応関係を示す図である。
【符号の説明】
１０１…文書登録サブシステム、
１０２…文書検索サーバ、
１０３…文書検索クライアント、
２１０…文書構造解析プログラム、
２１１…構造インデックス作成プログラム、
２１２…構造化全文データ生成プログラム、
２１３…文字列インデックス作成プログラム、
２１４…文書登録制御プログラム、
２１７…解析済み文書データ格納領域、

２１８…構造インデックス格納領域、
２１９…構造化全文データ格納領域、
２２０…文字列インデックス格納領域、
１７０１…検索条件解析プログラム、
１７０２…文字列インデックス検索プログラム、
１７０３…文書検索制御プログラム、
１７０４…検索結果データ格納領域、
２００１…検索条件の一例、
２００２…構造インデックスの一例、
２００３…展開済み検索条件データの一例、
２２０１…文字列インデックスの一例、
２５０１…検索条件入力プログラム、
２５０２…検索結果表示プログラム、
２５０３…クライアント制御プログラム、
２９０１…逆順構造インデックス作成プログラム、
２９０２…逆順構造インデックス格納領域、
４１０２…逆順構造インデックスの一例、
４４０１…文書構造正規化プログラム、
４８０１…メタ構造インデクス作成プログラム、
４８０２…メタ構造インデクス格納領域、
４８０３…種別定義テーブル、
４９０１…構造インデクスの一例、
４９０２…構造インデクスの一例、
４９０３…ルートメタノード、
４９０４…メタ構造インデクスの一例、
５５０１…メタ構造インデクス対応検索条件解析プログラム、
５８０１…検索条件の一例、
５８０２…メタ構造インデクスの一例、
５８０３…展開済み検索条件データの一例、
６００１…ルートノード付加プログラム、
６１０１…解析済み文書データの一例、
６１０２…ルートノード付加解析済み文書データの一例、
６２０１…検索条件修正プログラム、
６５０１…別名構造インデクスの例、
６５０２…メタ構造インデクスの例、
６５０３…別名定義テーブルの例、
６６０１…別名構造インデクス作成プログラム、
６６０２…別名構造インデクス格納領域、
６９０１…別名構造インデクス対応検索条件解析プログラム、
７３０１…メタ構造インデクスの例、
７３０２…ルートメタノード、
７３０３…最上位構造種別定義テーブル、
７３０４…構造インデクス１、
７３０５…種別定義テーブル１、
７３０６…構造インデクス２、
７３０７…種別定義テーブル２。

【図１】

図1



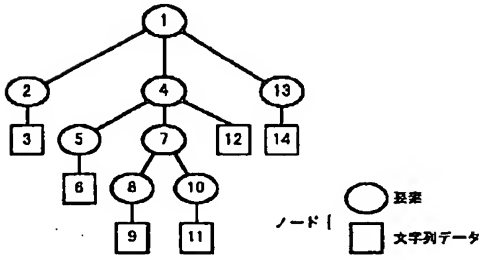
【図４】

図４

```
<!ELEMENT 論文          (タイトル，執筆者，日付，
                         本文，文献リスト)>

<!ELEMENT タイトル      (#PCDATA)>
<!ELEMENT 執筆者        (名前+)>
<!ELEMENT 名前          (#PCDATA)>
<!ELEMENT 日付          (#PCDATA)>

<!ELEMENT 本文          (章*)>
<!ELEMENT 章            (章題，(段落|備考)*，節*)>
<!ELEMENT 章題          (#PCDATA)>
<!ELEMENT 節            (節題，(段落|備考)*，項*)>
<!ELEMENT 節題          (#PCDATA)>
<!ELEMENT 項            (項題，(段落|備考)*)>
<!ELEMENT 項題          (#PCDATA)>
<!ELEMENT 段落          (#PCDATA|強調)*>
<!ELEMENT 備考          (#PCDATA|強調)*>
<!ATTLIST 備考
          type  (参考|注釈)  参考>
<!ELEMENT 強調          (#PCDATA)>

<!ELEMENT 文献リスト    (文献+)>
<!ELEMENT 文献      (タイトル，執筆者，出典)>
<!ELEMENT 出典          (#PCDATA)>
```

【図１０】

図１０



【図５０】

図５０

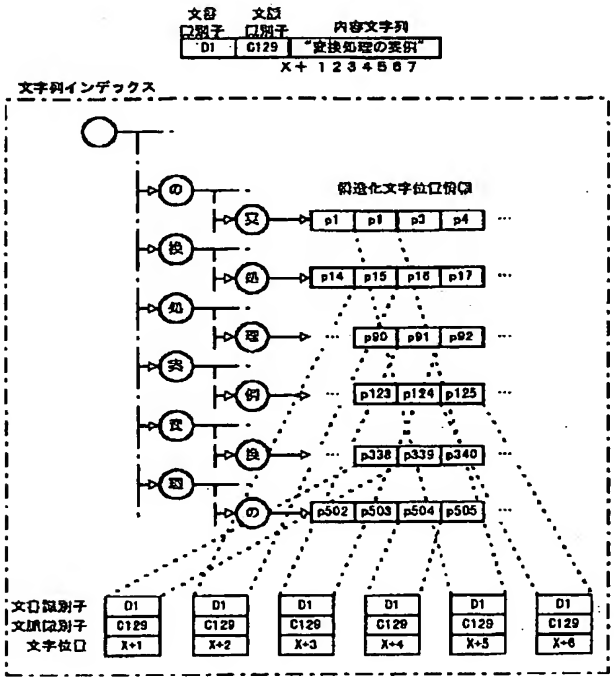
初別定義テーブル(４８０３)

初別	区分型名数	区分型名
図性	3	図性　登録　管理位置　…
日付	2	発行日　登録日
図目	4	タイトル　図目　…
:		

5001　5002　5003

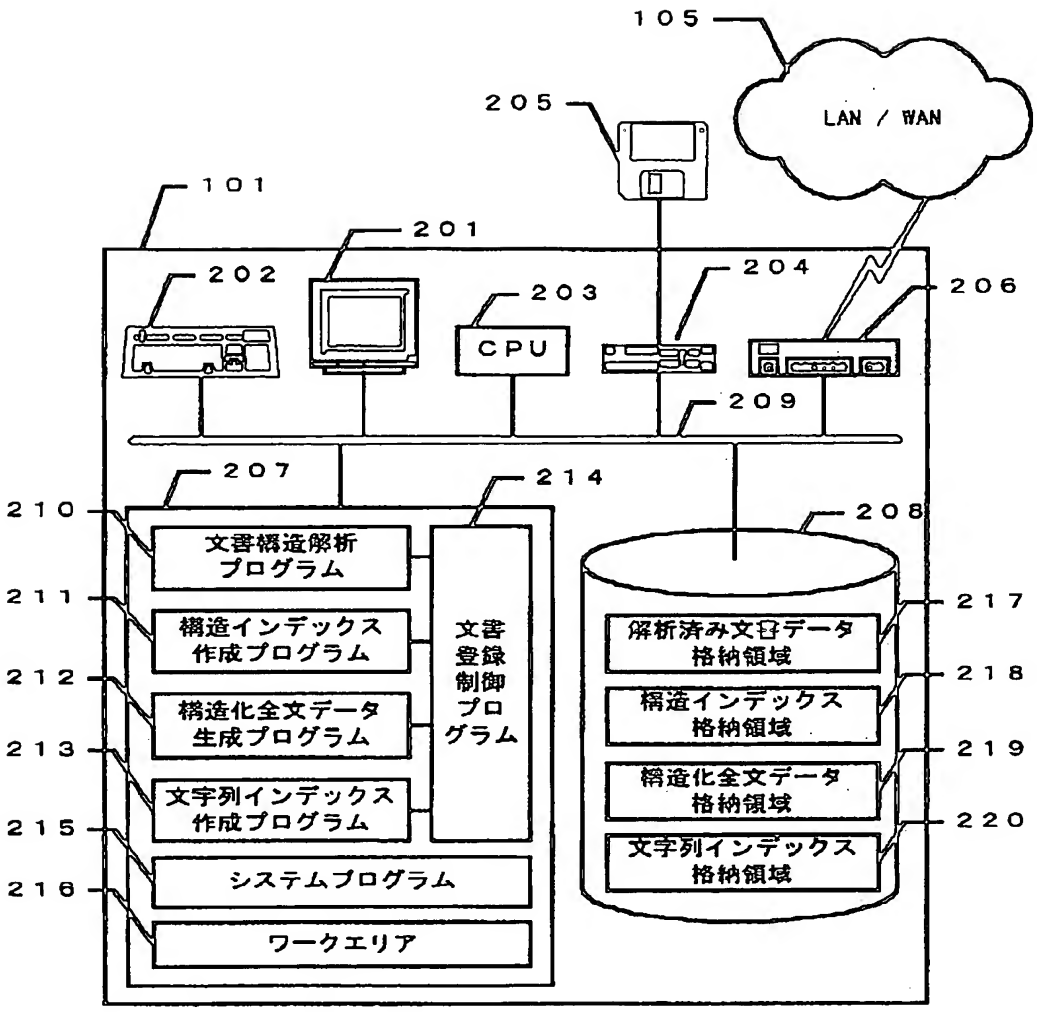
【図１６】

図１６



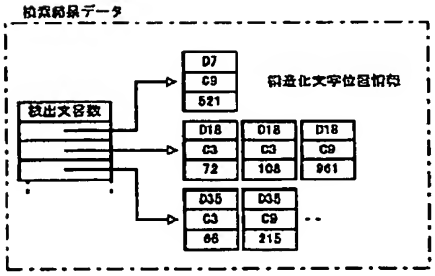
【図２】

図２


105
LAN / WAN
205
101
201
202
203
CPU
204
206
209
207
214
208
210
211
212
213
215
216
文書構造解析プログラム
構造インデックス作成プログラム
構造化全文データ生成プログラム
文字列インデックス作成プログラム
システムプログラム
ワークエリア
文書登録制御プログラム
解析済み文書データ格納領域
構造インデックス格納領域
構造化全文データ格納領域
文字列インデックス格納領域
217
218
219
220

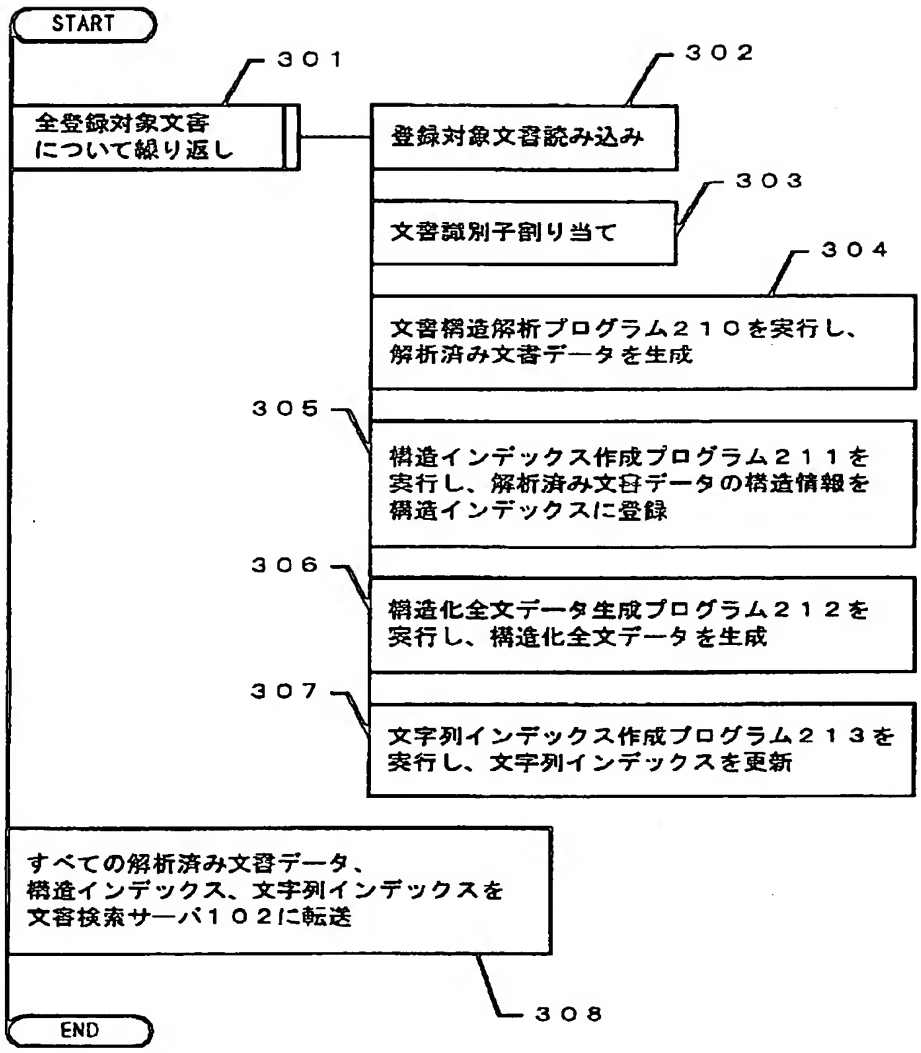
【図２３】

図２３


検索結果データ
検出文書数
構造化文字位置情報
D7
C9
521
D18
C3
72
108
961
D35
C9
215
66

【図３】

図３


START
３０１
全登録対象文書
について繰り返し
３０２
登録対象文書読み込み
３０３
文書識別子割り当て
３０４
文書構造解析プログラム２１０を実行し、
解析済み文書データを生成
３０５
構造インデックス作成プログラム２１１を
実行し、解析済み文書データの構造情報を
構造インデックスに登録
３０６
構造化全文データ生成プログラム２１２を
実行し、構造化全文データを生成
３０７
文字列インデックス作成プログラム２１３を
実行し、文字列インデックスを更新
すべての解析済み文書データ、
構造インデックス、文字列インデックスを
文書検索サーバ１０２に転送
３０８
END

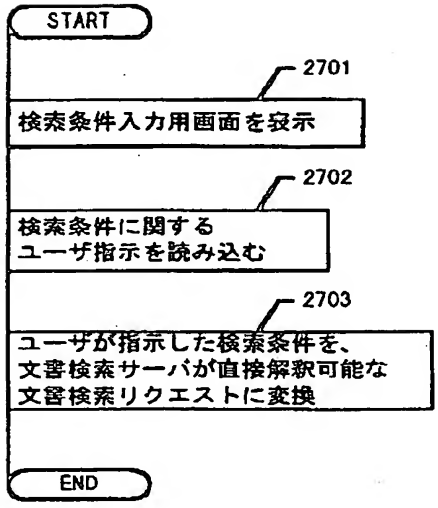
【図５】

図５

```
<!DOCTYPE 論文　SYSTEM "ronbun.dtd">
<論文>
<タイトル>SGML文書変換言語の開発とその適用事例</タイトル>
<執筆者>
<名前>高橋亨</名前>
<名前>東野純一</名前>
<名前>星幸雄</名前>
</執筆者>
<日付>1996年10月23日</日付>
<本文>
<章>
<章題>はじめに</章題>
<段落>文書記述にSGMLを用いることによって…</段落>
<段落>作成したSGML文書をさまざまな…</段落>…
</章>…
<章>
<章題>適用事例</章題>
<節>
<節題>背景</節題>
<段落>現在、ISOでは…</段落>…
</節>…
<節>
<節題>変換処理の実例</節題>
<項>
<項題>数式の変換</項題>
<段落>JIS規格DTDでは、基本的には数式を…</段落>…
<備考 type=注釈>ただし、行列式の場合には…</備考>…
</項>…
</節>
</章>
</本文>
<文献リスト>
<文献>
<タイトル>SGMLインスタンスの変換方式の検討</タイトル>
<執筆者>
<名前>今郷誼</名前>…
</執筆者>
<出典>情報処理学会第49回全国大会</出典>
</文献>…
</文献リスト>
</論文>
```

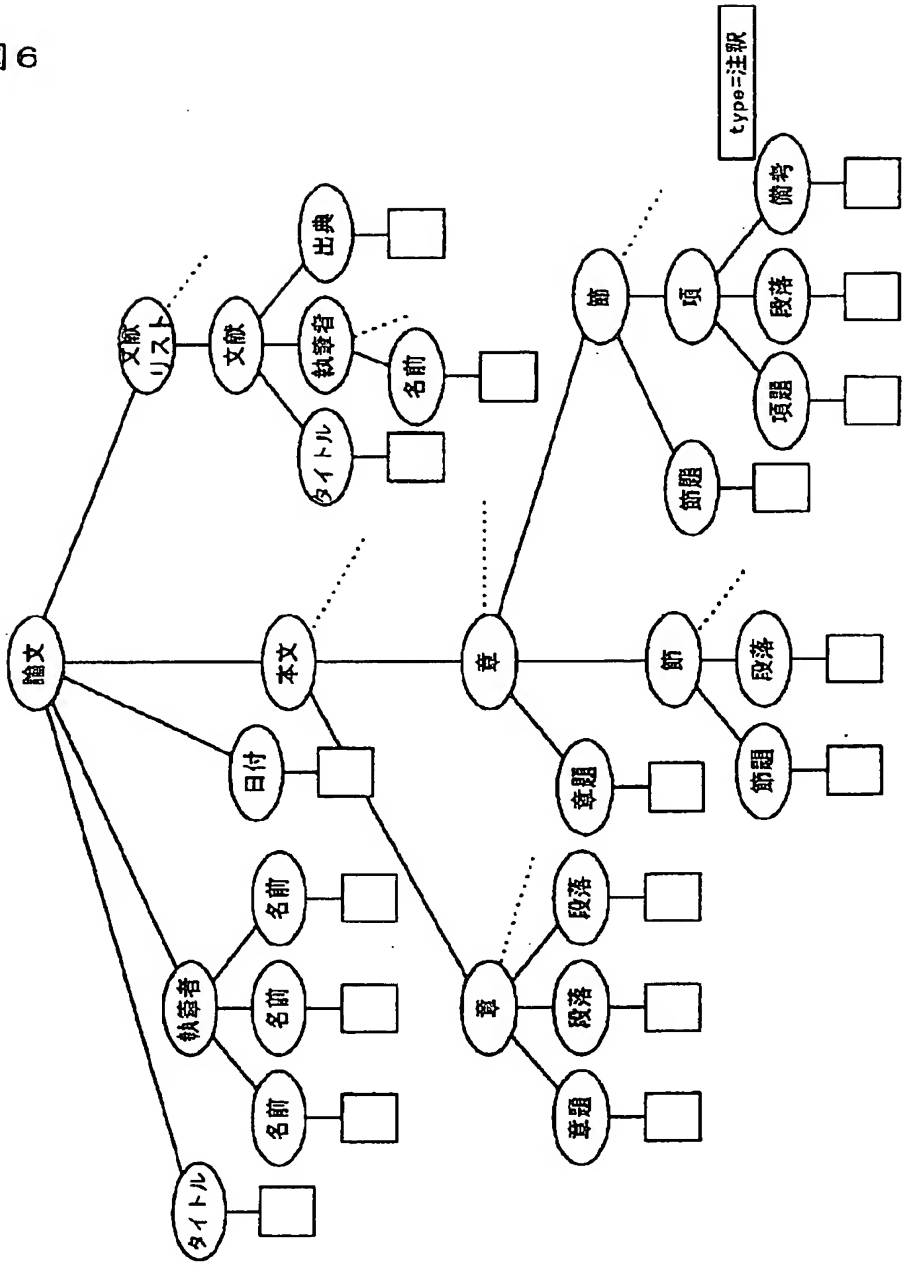
【図２７】

図２７



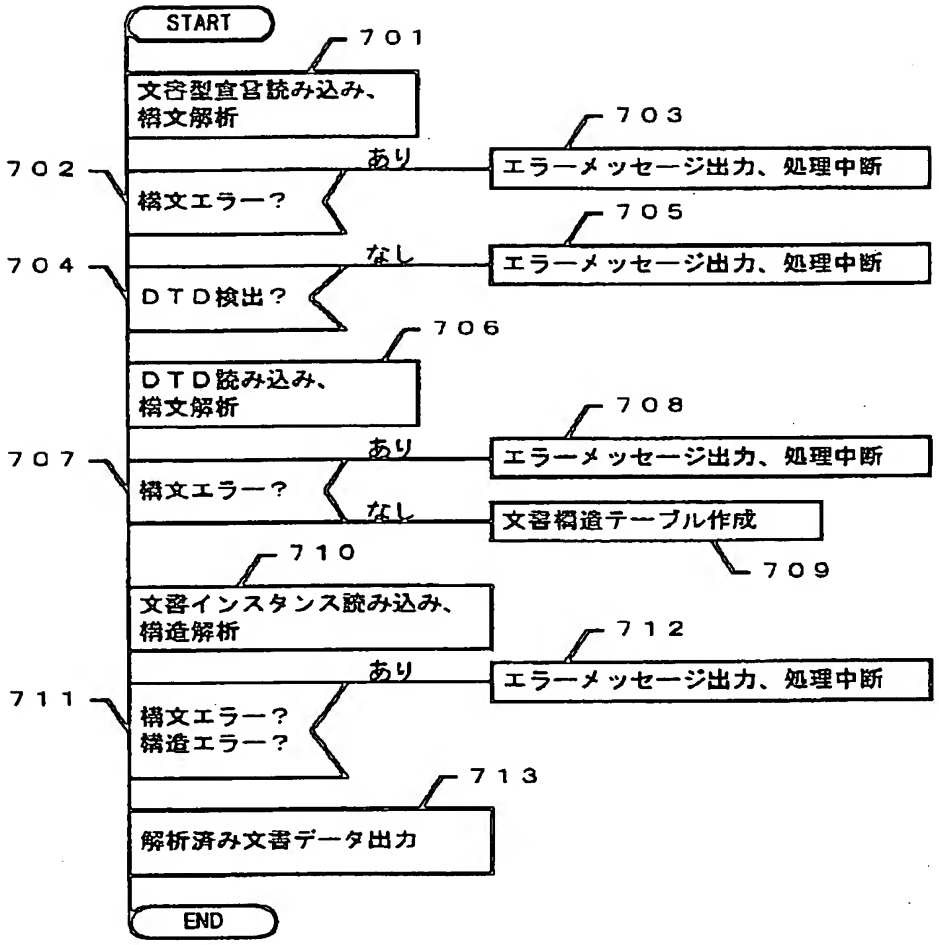
【図６】

図６


論文
本文
日付
執筆者
名前
名前
名前
タイトル
文献リスト
文献
出典
執筆者
名前
タイトル
章
節
項
備考
段落
項題
節題
type=注釈
章
節
段落
節題
章題
章
段落
段落
章題

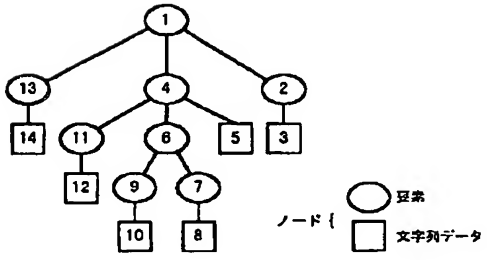
【図７】

図７



【図３２】

図３２



【図７２】

図７２

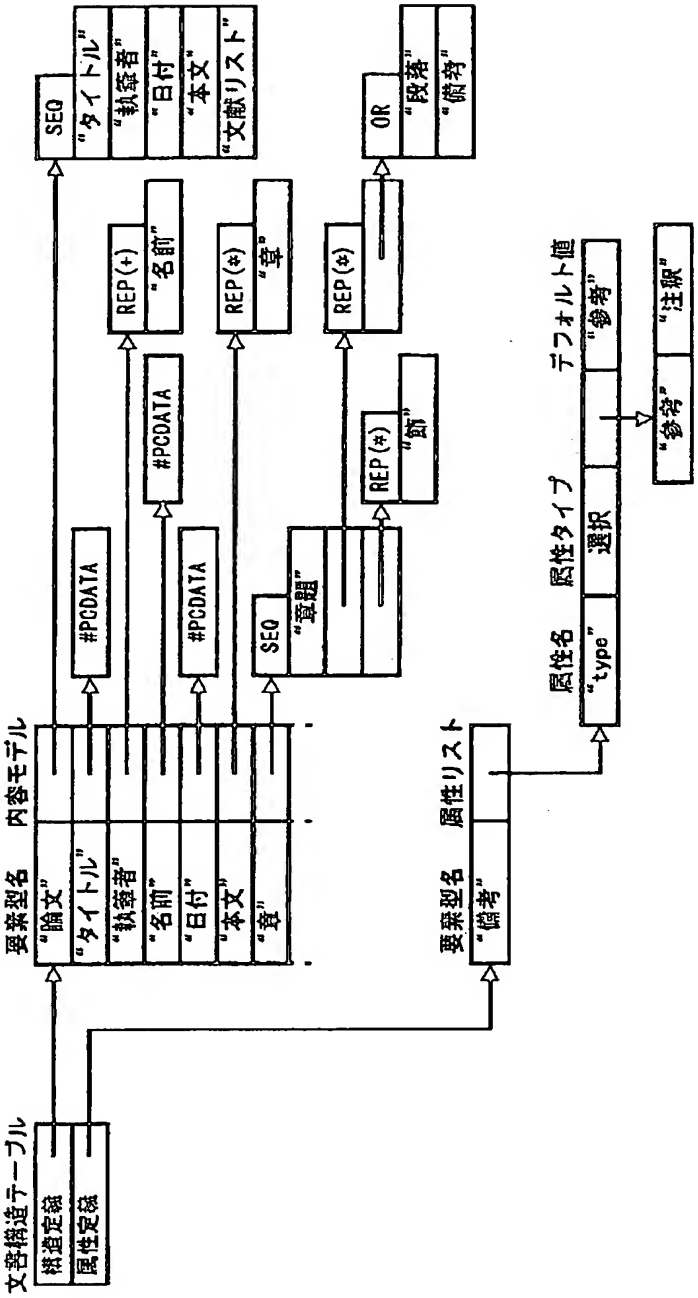
段別定義テーブル(４８０３)

段別	段構造数	ＤＴＤ名称および段構造名
属性	3	文書Ｄ.dtd:Ｄ属性　節文.dtd:Ｄ段　…
段Ｄ内容	1	文書Ｄ.dtd:本文
本文	3	節文.dtd:本文　…
:		

5001　5002　7201

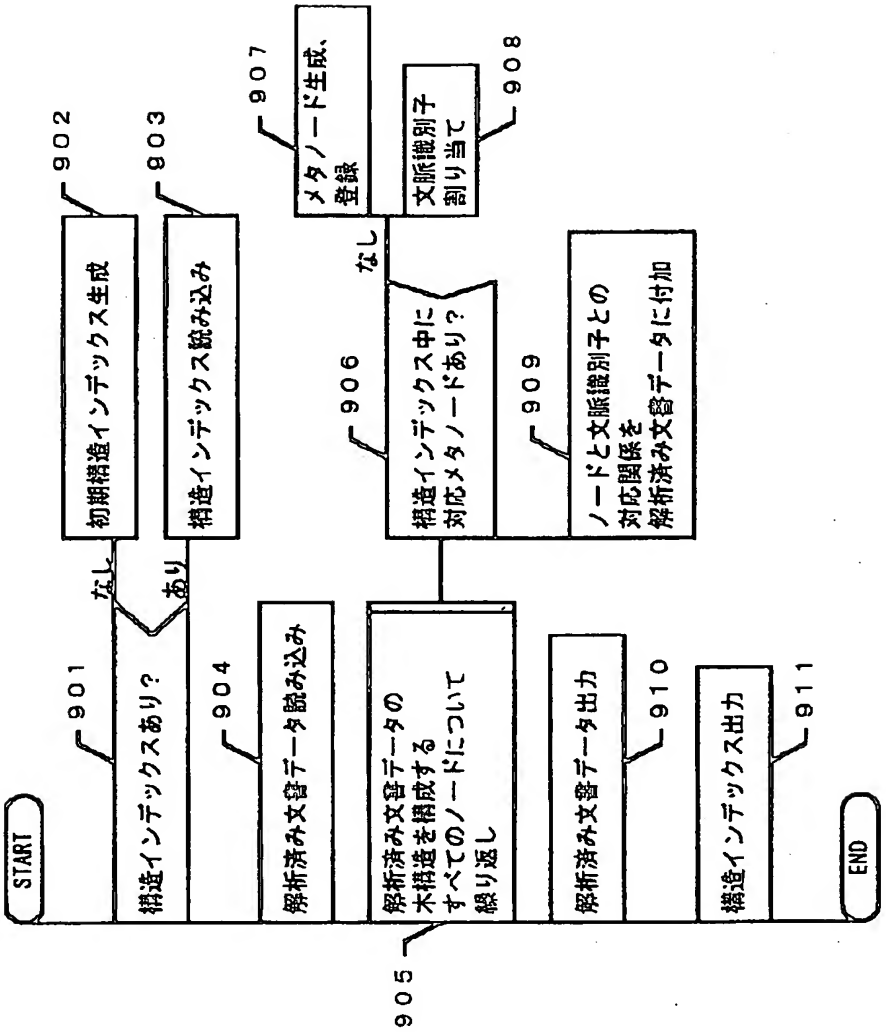
【図８】

図８


文容構造テーブル
構造定義
属性定義
要素型名
内容モデル
"論文"
"タイトル"
"執筆者"
"名前"
"日付"
"本文"
"章"
SEQ
"タイトル"
"執筆者"
"日付"
"本文"
"文献リスト"
REP(+)
"名前"
REP(*)
"章"
SEQ
"章題"
REP(*)
"節"
OR
"段落"
"備考"
#PCDATA
#PCDATA
#PCDATA
要素型名
属性リスト
"備考"
属性名
属性タイプ
デフォルト値
"type"
選択
"参考"
"注釈"
"参考"

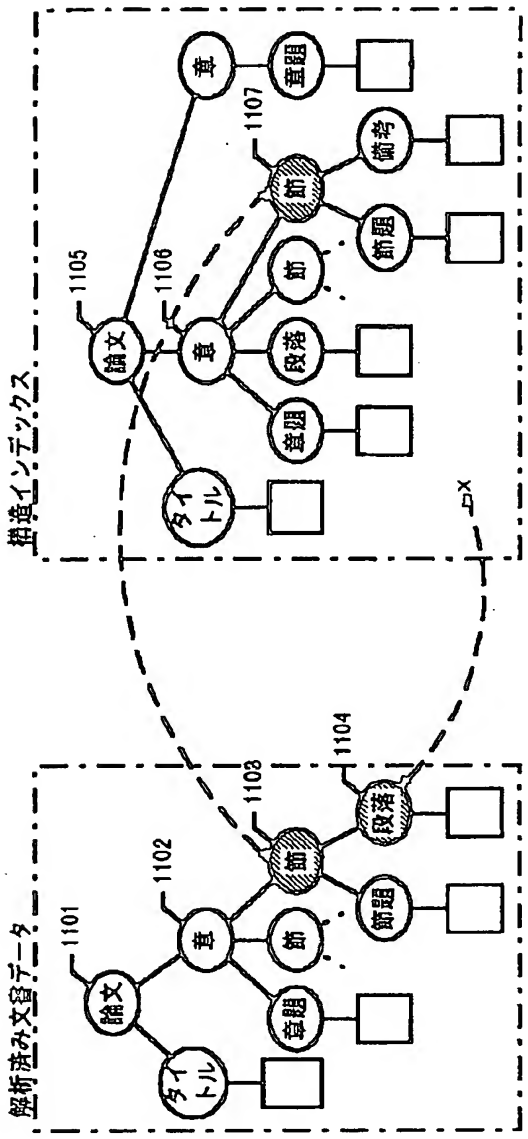
【図９】

図９


START
901
構造インデックスあり？
なし
あり
902
初期構造インデックス生成
903
構造インデックス読み込み
904
解析済み文書データ読み込み
905
解析済み文書データの
木構造を構成する
すべてのノードについて
繰り返し
906
構造インデックス中に
対応メタノードあり？
なし
907
メタノード生成、
登録
908
文脈識別子
割り当て
909
ノードと文脈識別子との
対応関係を
解析済み文書データに付加
910
解析済み文書データ出力
911
構造インデックス出力
END

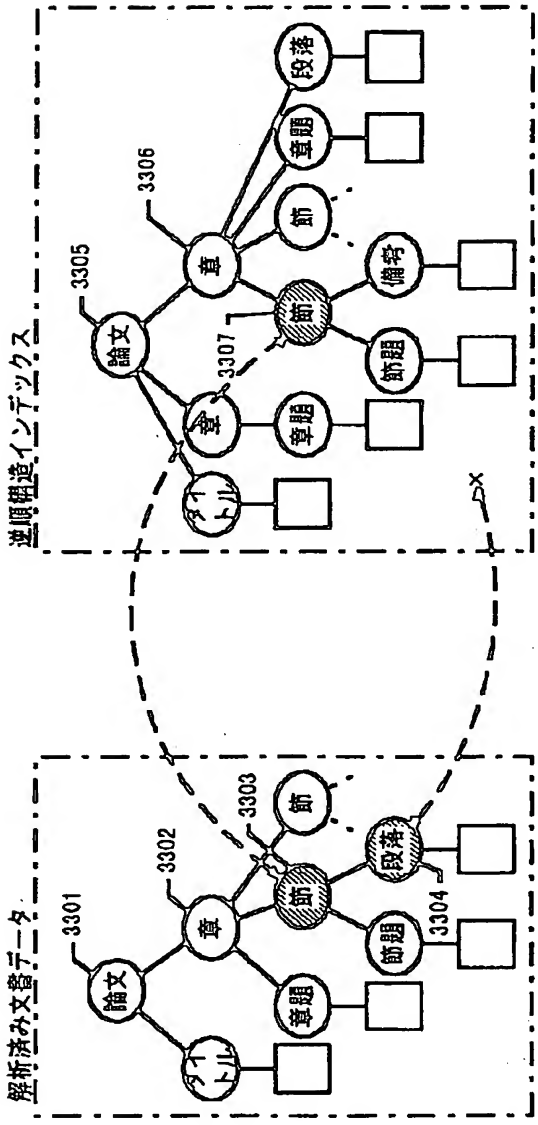
【図１１】

図１１



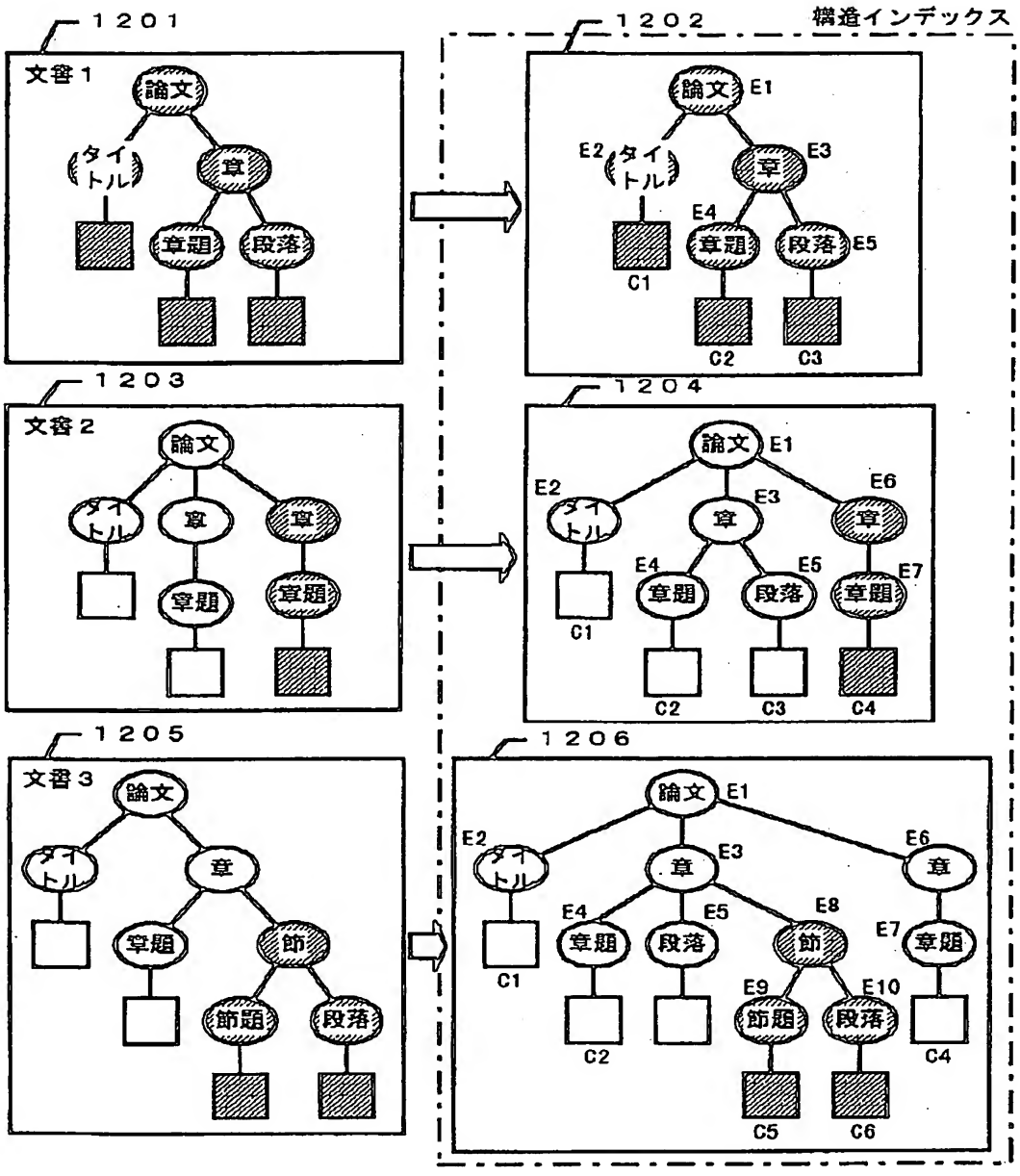
【図３３】

図３３



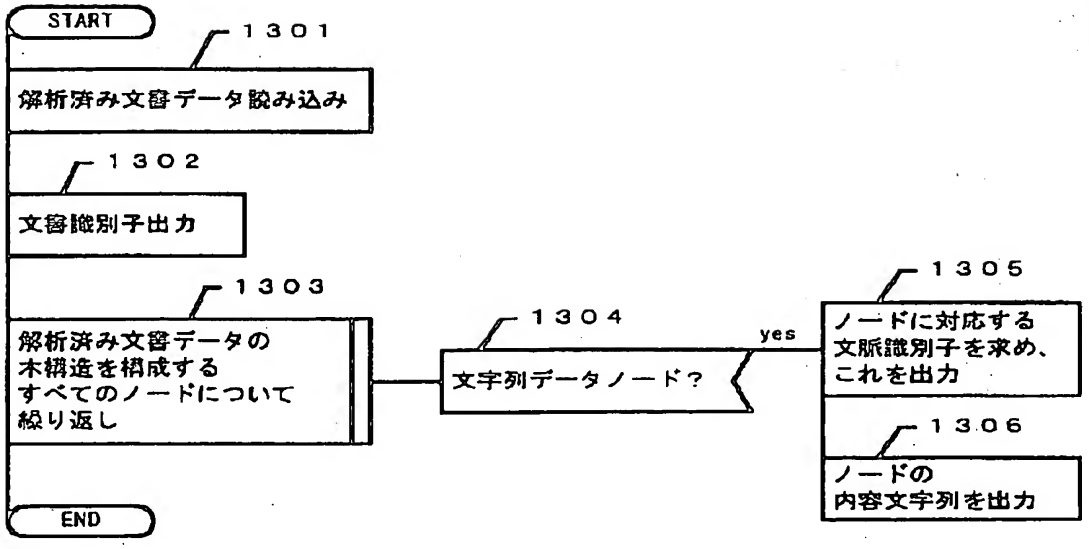
【図１２】

図１２


構造インデックス
１２０１
文書１
論文
タイトル
章
章題
段落
１２０２
論文 E1
E2 タイトル
章 E3
E4 章題
段落 E5
C1
C2
C3
１２０３
文書２
論文
タイトル
章
章
章題
章題
１２０４
論文 E1
E2 タイトル
E3 章
E6 章
E4 章題
段落 E5
E7 章題
C1
C2
C3
C4
１２０５
文書３
論文
タイトル
章
章題
節
節題
段落
１２０６
論文 E1
E2 タイトル
E3 章
E6 章
E4 章題
段落 E5
E8 節
E7 章題
C1
C2
E9 節題
E10 段落
C4
C5
C6

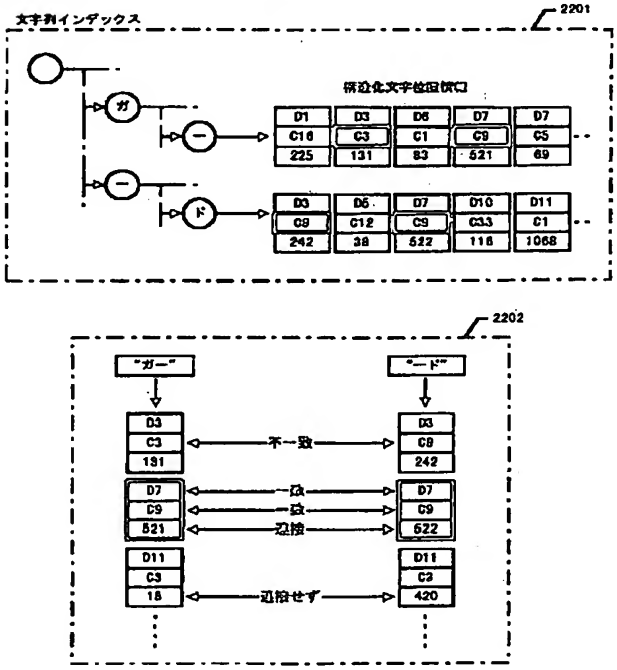
【図１３】

図１３



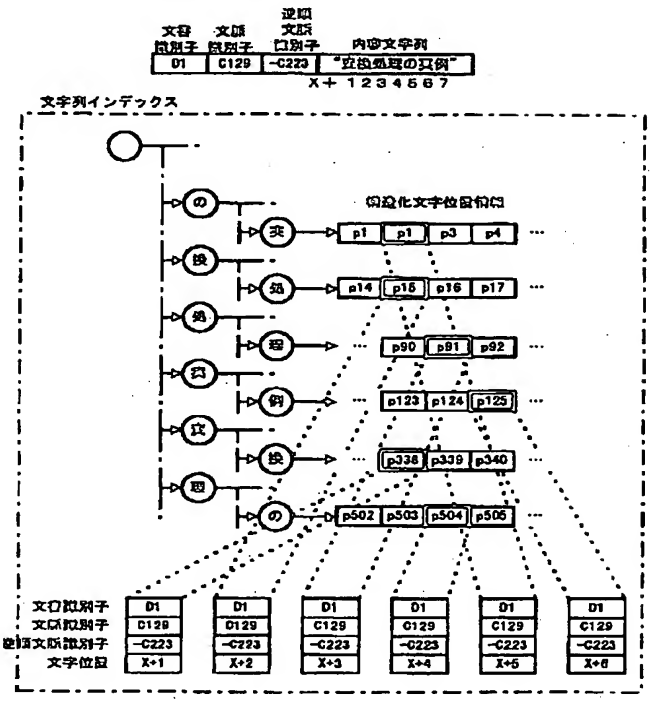
【図２２】

図２２



【図３７】

図３７



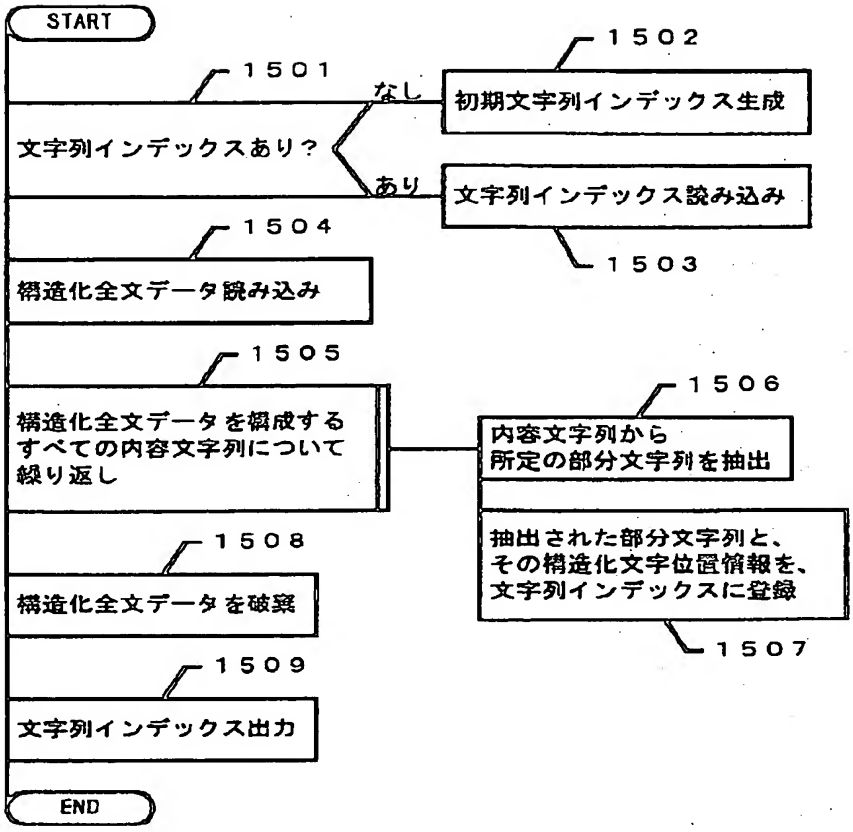
【図１４】

図１４

文書識別子	文脈識別子	内容文字列
D1	C1	“SGML文書変換言語の開発とその適用事例”
	C2	“高橋亨”
	C3	“東野純一”
	C4	“星幸雄”
	C5	“1996年10月23日”
	C6	“はじめに”
	C7	“文書記述にSGMLを用いることによって…”
	C8	“作成したSGML文書をさまざまな…”
	⋮	⋮
	C15	“適用事例”
	C16	“背景”
	C17	“現在、ISOでは…”
	⋮	⋮
	C129	“変換処理の実例”
	C130	“数式の変換”
	C131	“JIS規格DTDでは、基本的には数式を…”
	C132	“ただし、行列式の場合には…”
	⋮	⋮

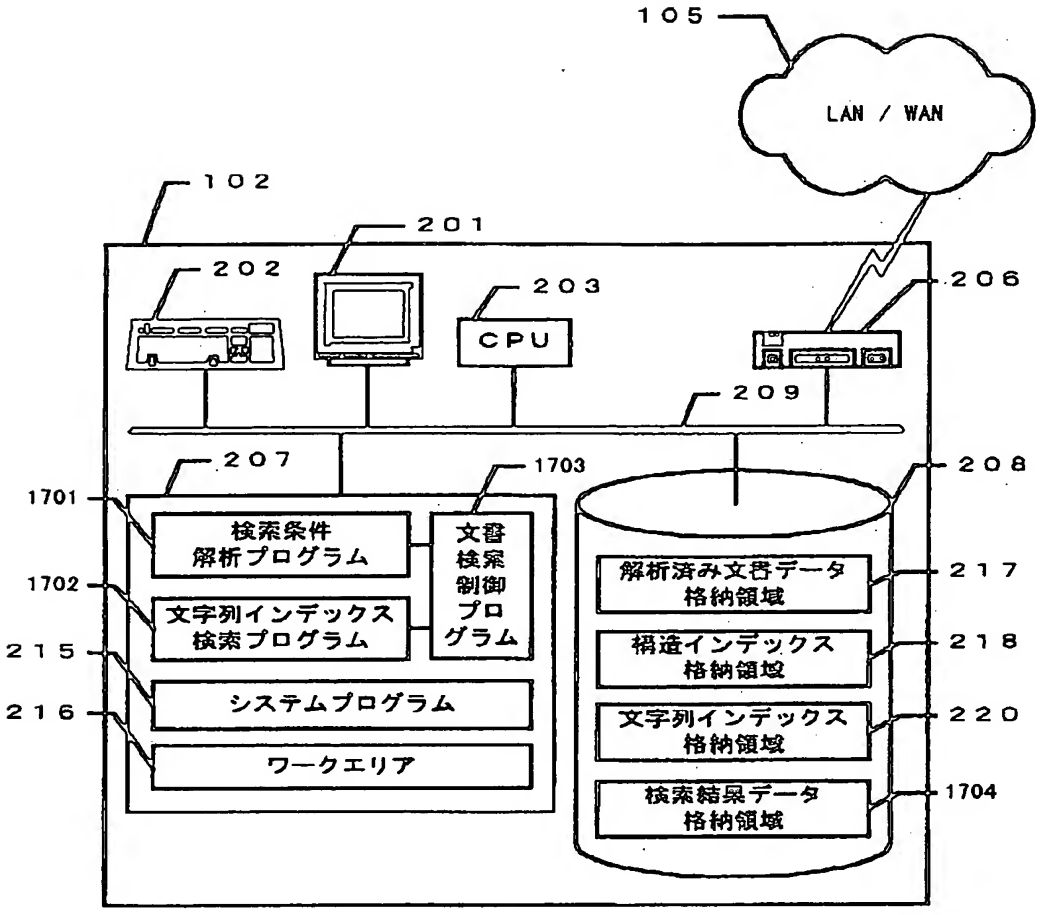
【図１５】

図１５


START
１５０１
文字列インデックスあり？
なし
あり
１５０２
初期文字列インデックス生成
文字列インデックス読み込み
１５０３
１５０４
構造化全文データ読み込み
１５０５
構造化全文データを構成する
すべての内容文字列について
繰り返し
１５０６
内容文字列から
所定の部分文字列を抽出
抽出された部分文字列と、
その構造化文字位置情報を、
文字列インデックスに登録
１５０７
１５０８
構造化全文データを破棄
１５０９
文字列インデックス出力
END

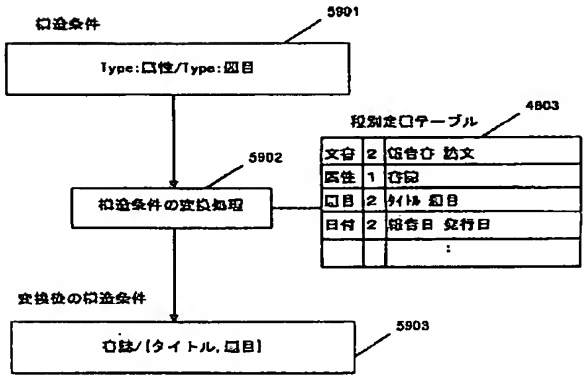
【図１７】

図１７



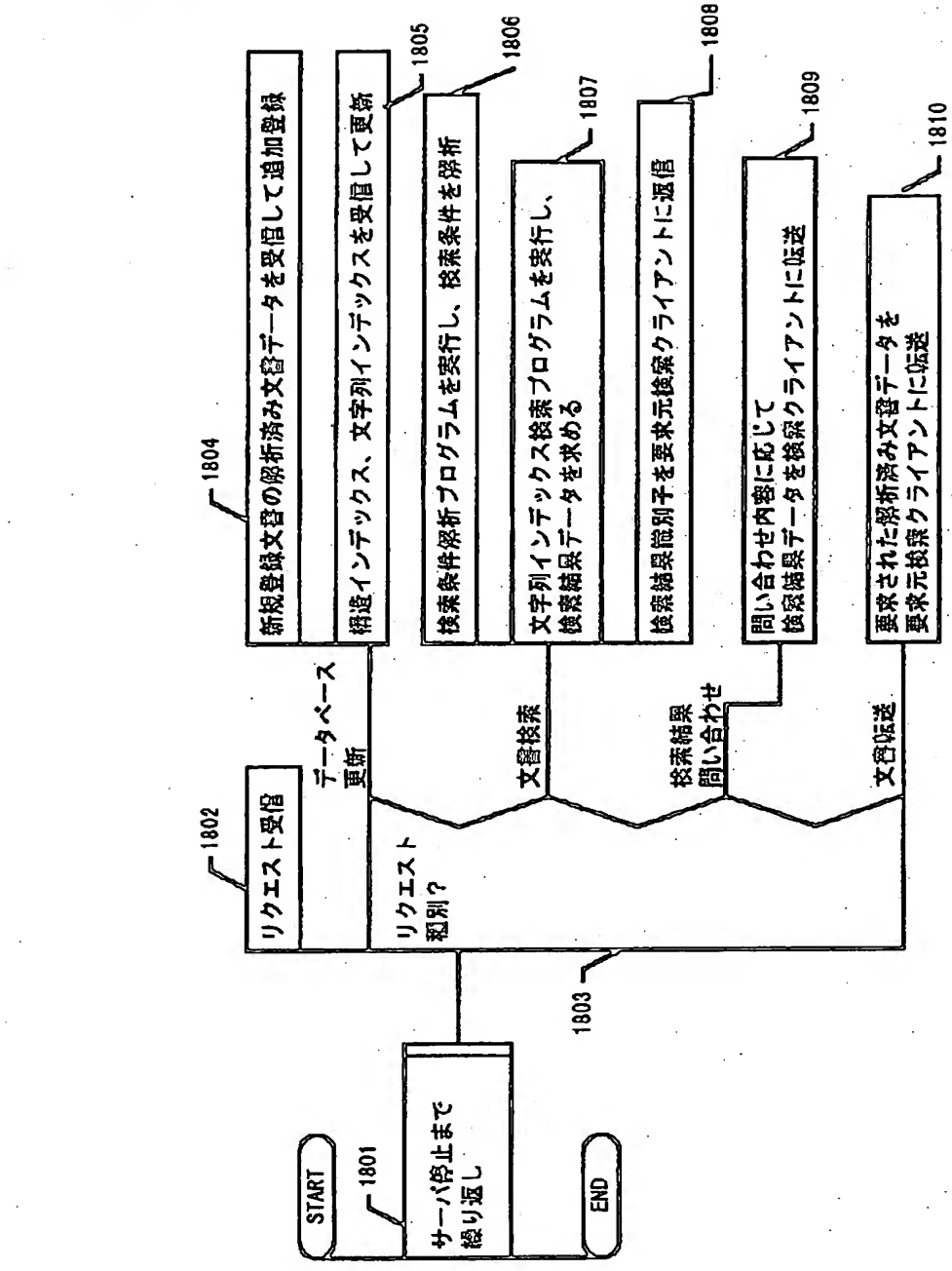
【図５９】

図５９



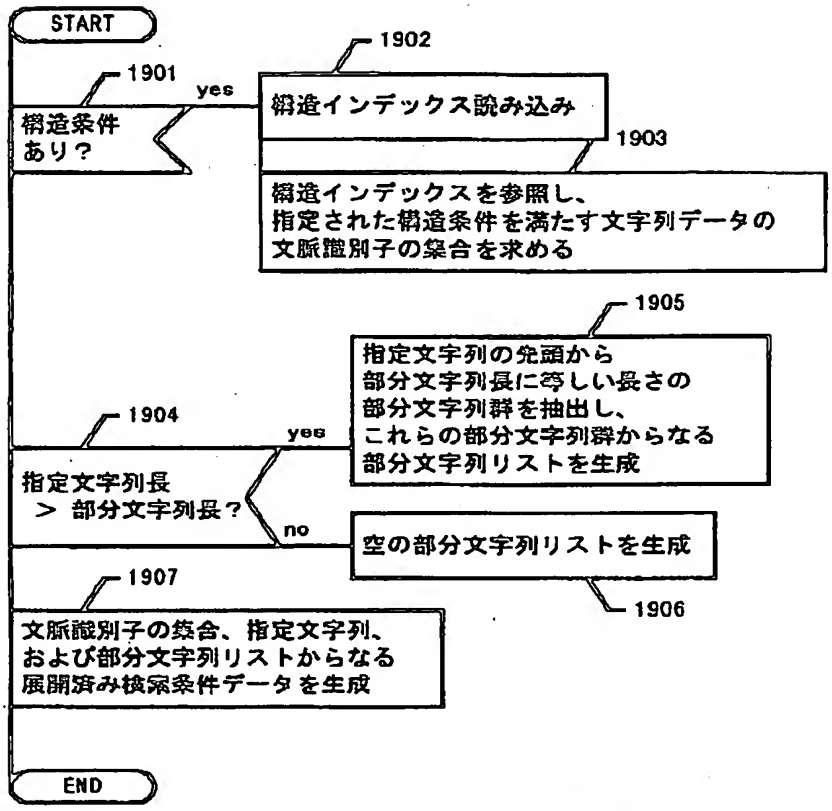
【図１８】

図１８


START
1801
サーバ停止まで
繰り返し
END
1803
1802
リクエスト受信
リクエスト
種別？
データベース
更新
文書検索
検索結果
問い合わせ
文書転送
1804
新規登録文書の解析済み文書データを受信して追加登録
1805
構造インデックス、文字列インデックスを受信して更新
1806
検索条件解析プログラムを実行し、検索条件を解析
1807
文字列インデックス検索プログラムを実行し、
検索結果データを求める
1808
検索結果識別子を要求元検索クライアントに返信
1809
問い合わせ内容に応じて
検索結果データを検索クライアントに返送
1810
要求された解析済み文書データを
要求元検索クライアントに返送

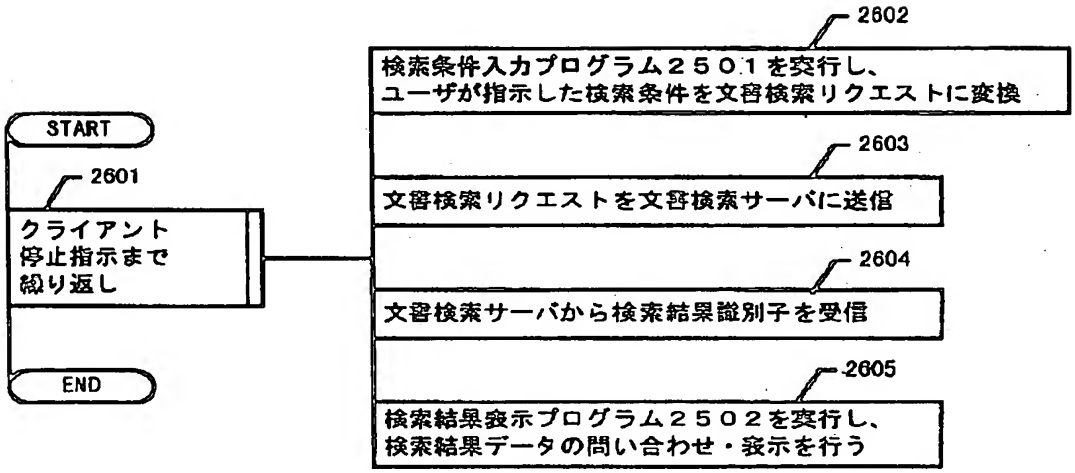
【図１９】

図１９



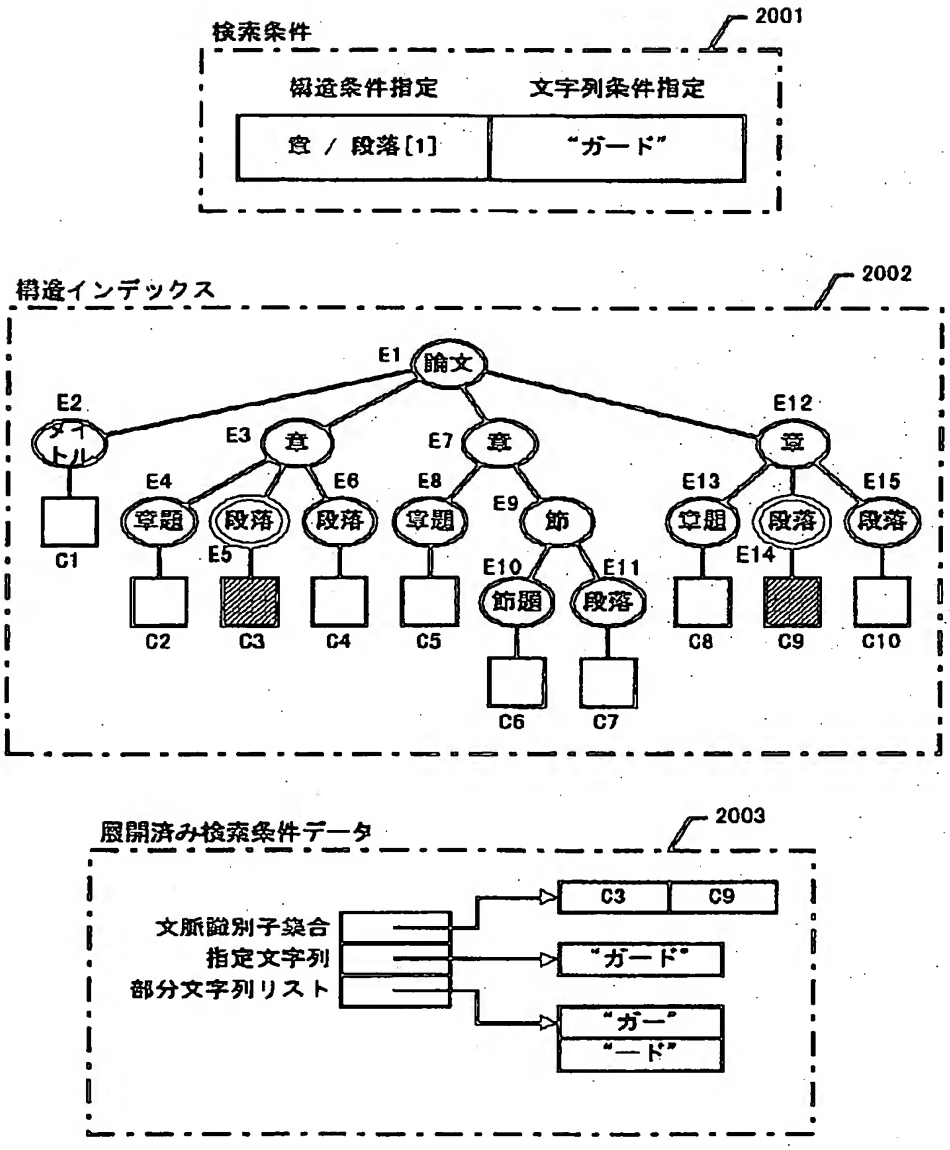
【図２６】

図２６



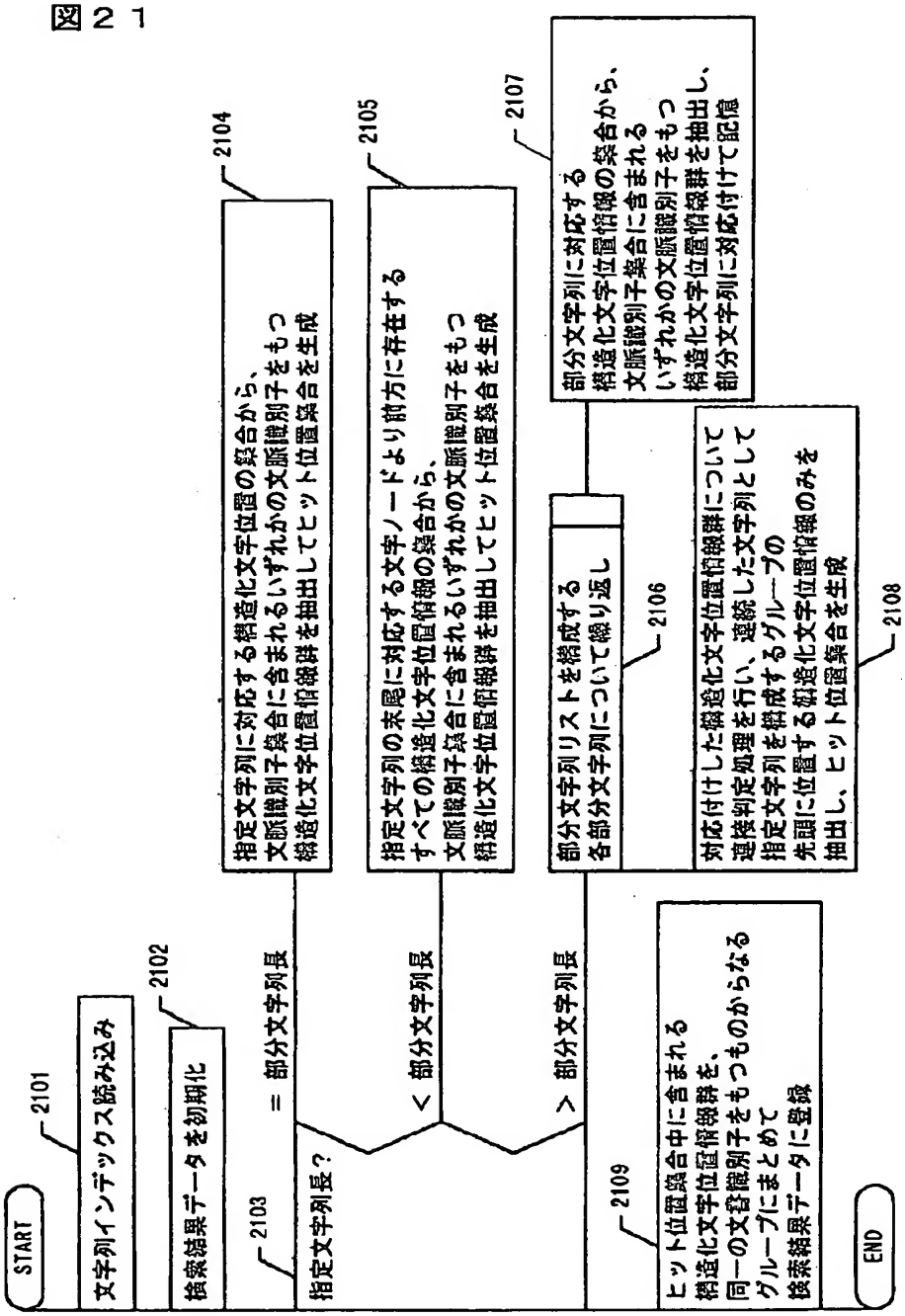
【図２０】

図２０


検索条件
2001
構造条件指定
文字列条件指定
章 / 段落[1]
"ガード"
構造インデックス
2002
E1
論文
E2
タイトル
E3
章
E7
章
E12
章
E4
章題
E5
段落
E6
段落
E8
章題
E9
節
E10
節題
E11
段落
E13
章題
E14
段落
E15
段落
C1
C2
C3
C4
C5
C6
C7
C8
C9
C10
展開済み検索条件データ
2003
文脈識別子集合
指定文字列
部分文字列リスト
C3
C9
"ガード"
"ガー"
"ード"

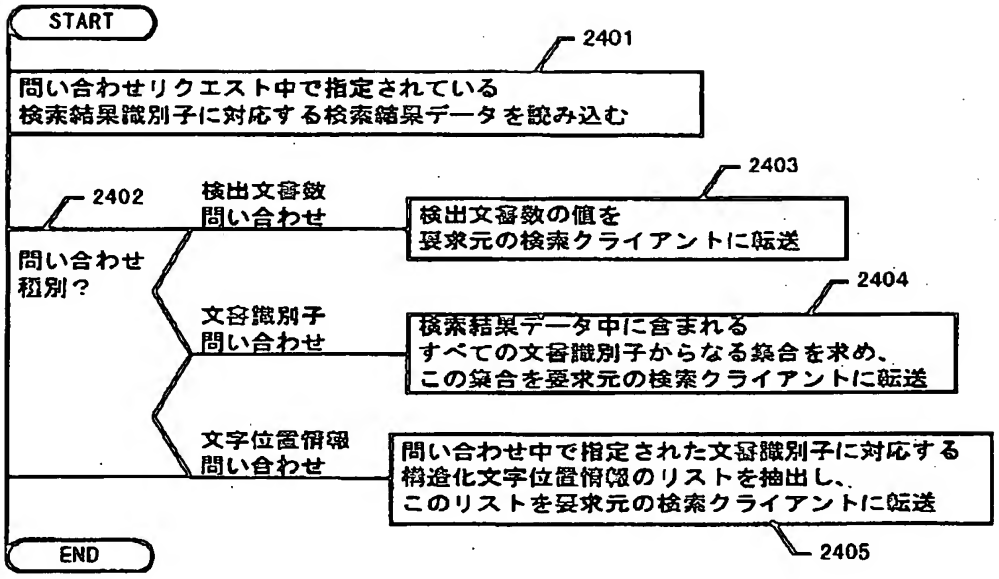
【図２１】

図２１


START
2101
文字列インデックス読み込み
2102
検索結果データを初期化
2103
指定文字列長?
= 部分文字列長
2104
指定文字列に対応する構造化文字位置の集合から、
文脈識別子集合に含まれるいずれかの文脈識別子をもつ
構造化文字位置情報群を抽出してヒット位置集合を生成
< 部分文字列長
2105
指定文字列の末尾に対応する文字ノードより前方に存在する
すべての構造化文字位置情報の集合から、
文脈識別子集合に含まれるいずれかの文脈識別子をもつ
構造化文字位置情報を抽出してヒット位置集合を生成
> 部分文字列長
2106
部分文字列リストを構成する
各部分文字列について繰り返し
2107
部分文字列に対応する
構造化文字位置情報の集合から、
文脈識別子集合に含まれる
いずれかの文脈識別子をもつ
構造化文字位置情報群を抽出し、
部分文字列に対応付けて記憶
2108
対応付けした構造化文字位置情報群について
連接判定処理を行い、連続した文字列として
指定文字列を構成するグループの
先頭に位置する構造化文字位置情報のみを
抽出し、ヒット位置集合を生成
2109
ヒット位置集合中に含まれる
構造化文字位置情報群を、
同一の文脈識別子をもつものからなる
グループにまとめて
検索結果データに登録
END

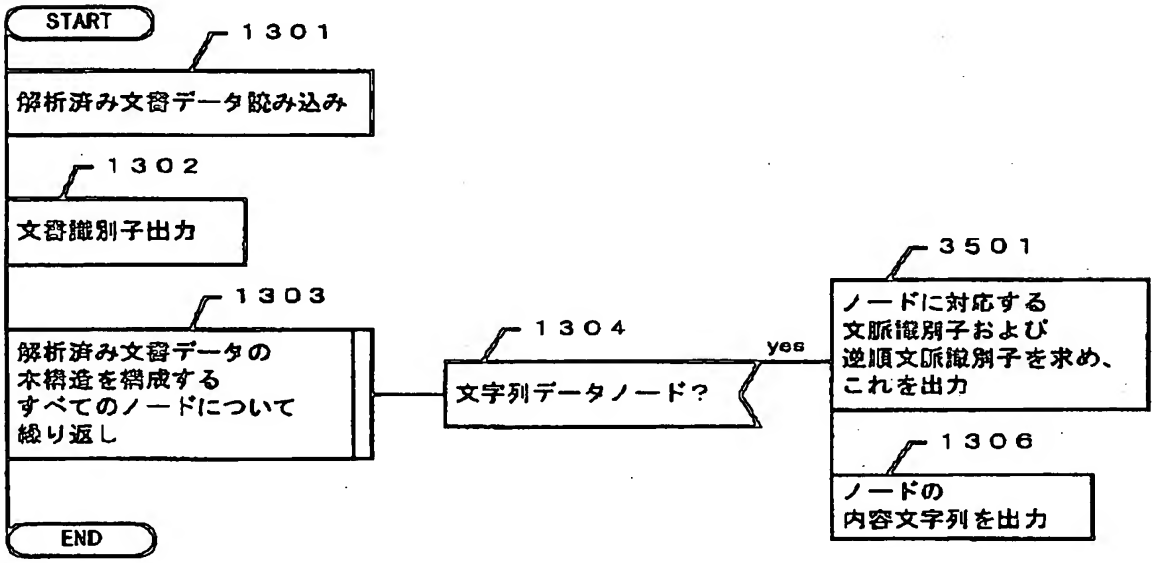
【図２４】

図２４



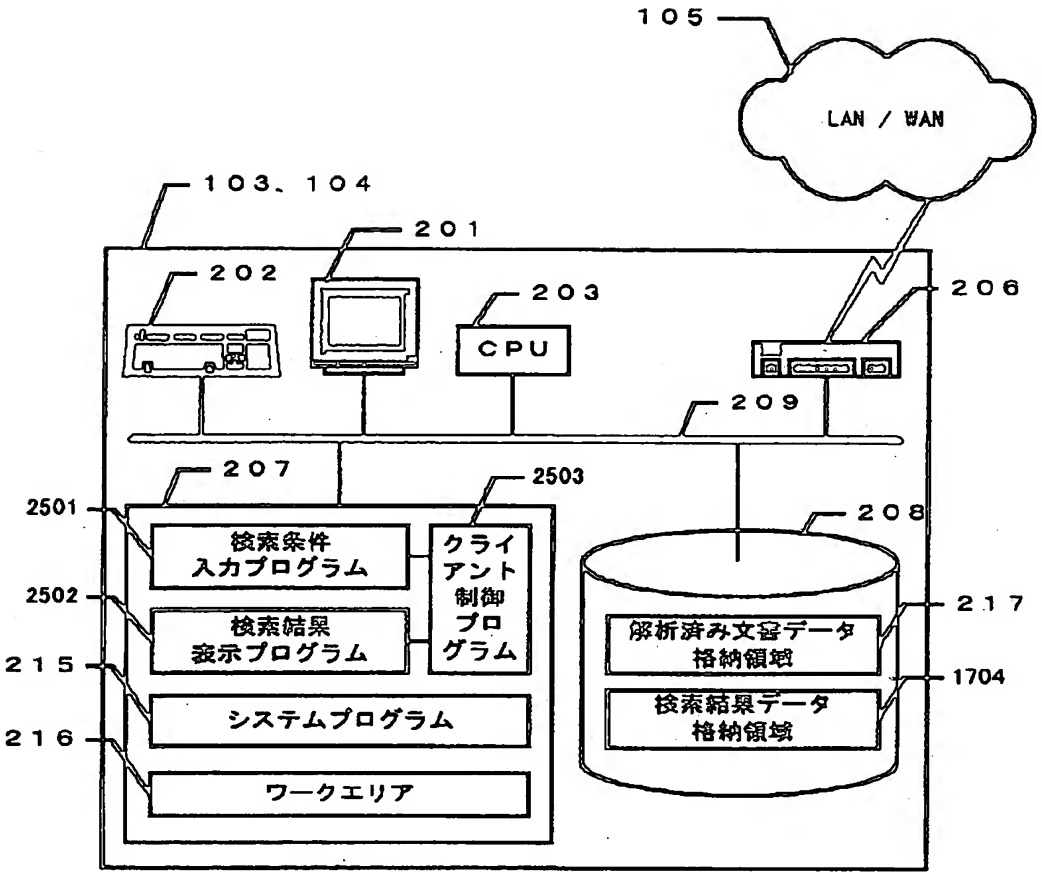
【図３５】

図３５



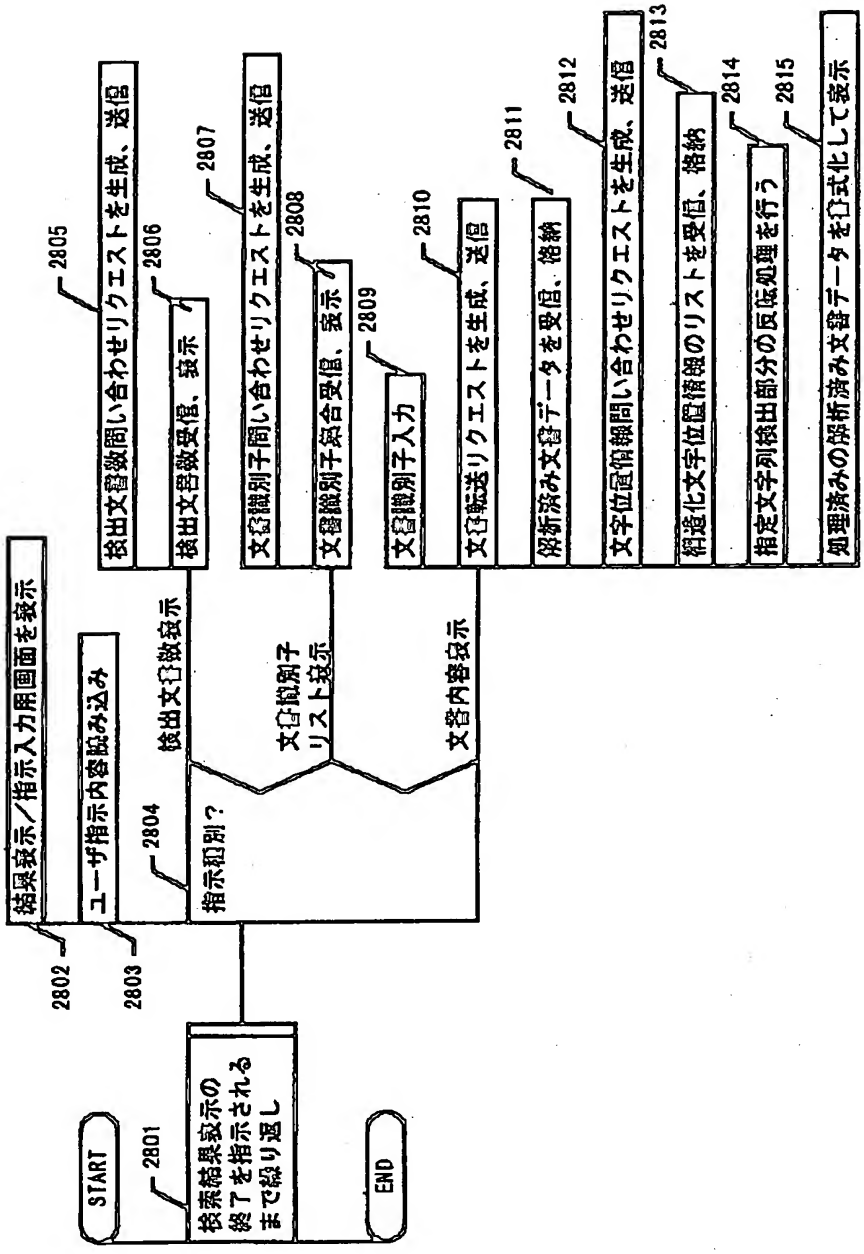
【図２５】

図２５


105
LAN / WAN
103、104
201
202
203
CPU
206
209
207
2503
2501
検索条件
入力プログラム
クライアント
制御
プログラム
2502
検索結果
表示プログラム
215
システムプログラム
216
ワークエリア
208
解析済み文書データ
格納領域
217
検索結果データ
格納領域
1704

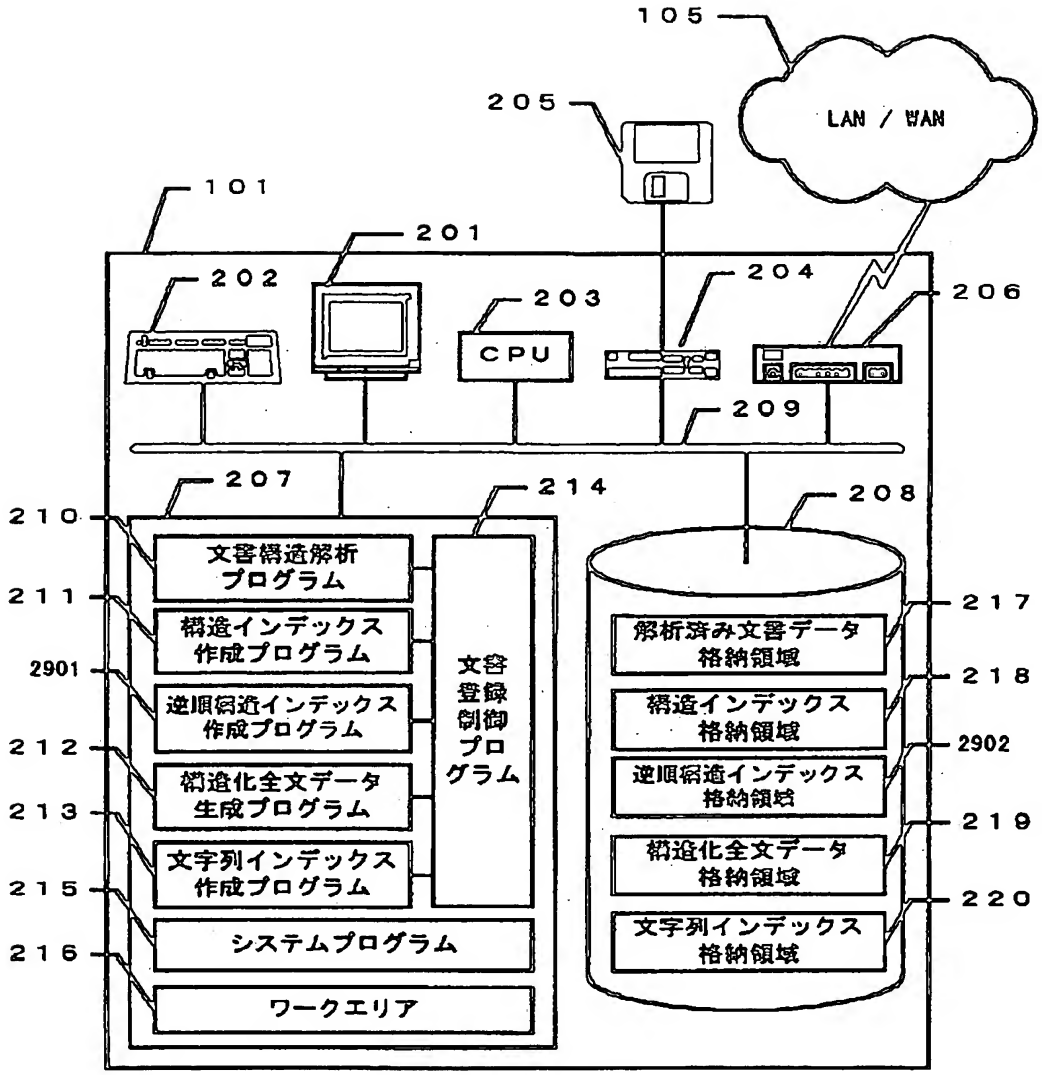
【図２８】

図２８


START
2801
検索結果表示の終了を指示されるまで繰り返し
END
2802
結果表示／指示入力用画面を表示
2803
ユーザ指示内容読み込み
2804
指示和則？
検出文書数表示
文書識別子リスト表示
文書内容表示
2805
検出文書数問い合わせリクエストを生成、送信
2806
検出文書数受信、表示
2807
文書識別子問い合わせリクエストを生成、送信
2808
文書識別子集合受信、表示
2809
文書識別子入力
2810
文書転送リクエストを生成、送信
2811
解析済み文書データを受信、格納
2812
文字位置情報問い合わせリクエストを生成、送信
2813
2814
2815
処理済みの解析済み文書データを汎式化して表示

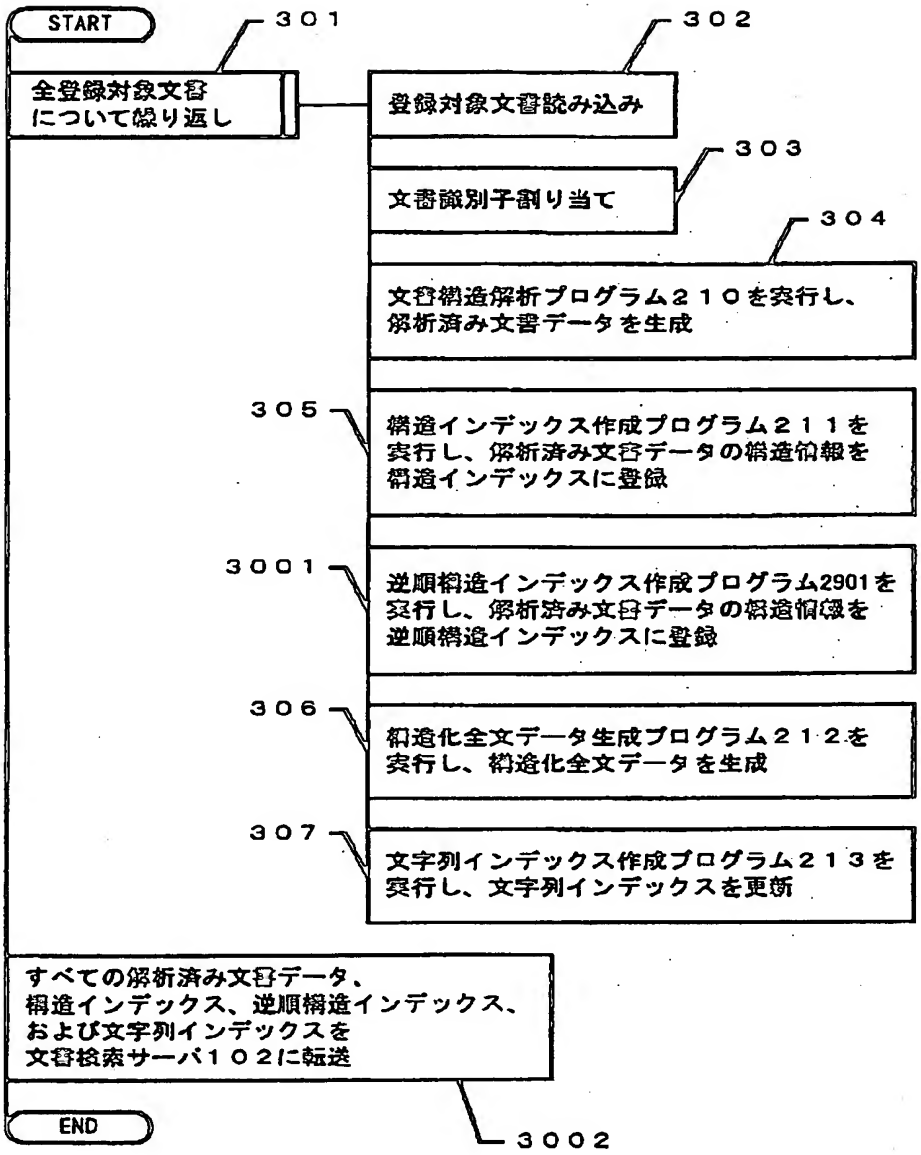
【図２９】

図２９


105
LAN / WAN
205
101
201
202
203
CPU
204
206
209
207
214
208
210
211
2901
212
213
215
216
文書構造解析プログラム
構造インデックス作成プログラム
逆順構造インデックス作成プログラム
構造化全文データ生成プログラム
文字列インデックス作成プログラム
システムプログラム
ワークエリア
文書登録制御プログラム
217
218
2902
219
220
解析済み文書データ格納領域
構造インデックス格納領域
逆順構造インデックス格納領域
構造化全文データ格納領域
文字列インデックス格納領域

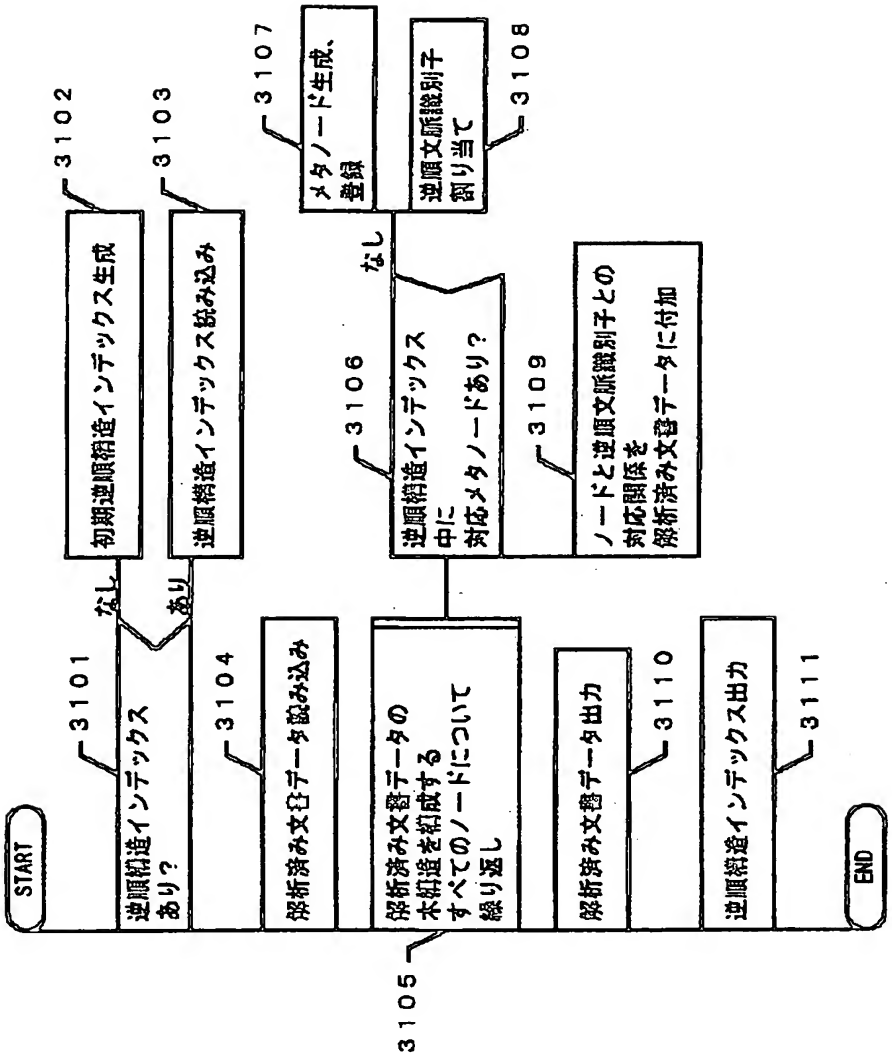
【図３０】

図３０


START
３０１
全登録対象文書
について繰り返し
３０２
登録対象文書読み込み
３０３
文書識別子割り当て
３０４
文書構造解析プログラム２１０を実行し、
解析済み文書データを生成
３０５
構造インデックス作成プログラム２１１を
実行し、解析済み文書データの構造情報を
構造インデックスに登録
３００１
逆順構造インデックス作成プログラム2901を
実行し、解析済み文書データの構造情報を
逆順構造インデックスに登録
３０６
構造化全文データ生成プログラム２１２を
実行し、構造化全文データを生成
３０７
文字列インデックス作成プログラム２１３を
実行し、文字列インデックスを更新
すべての解析済み文書データ、
構造インデックス、逆順構造インデックス、
および文字列インデックスを
文書検索サーバ１０２に転送
３００２
END

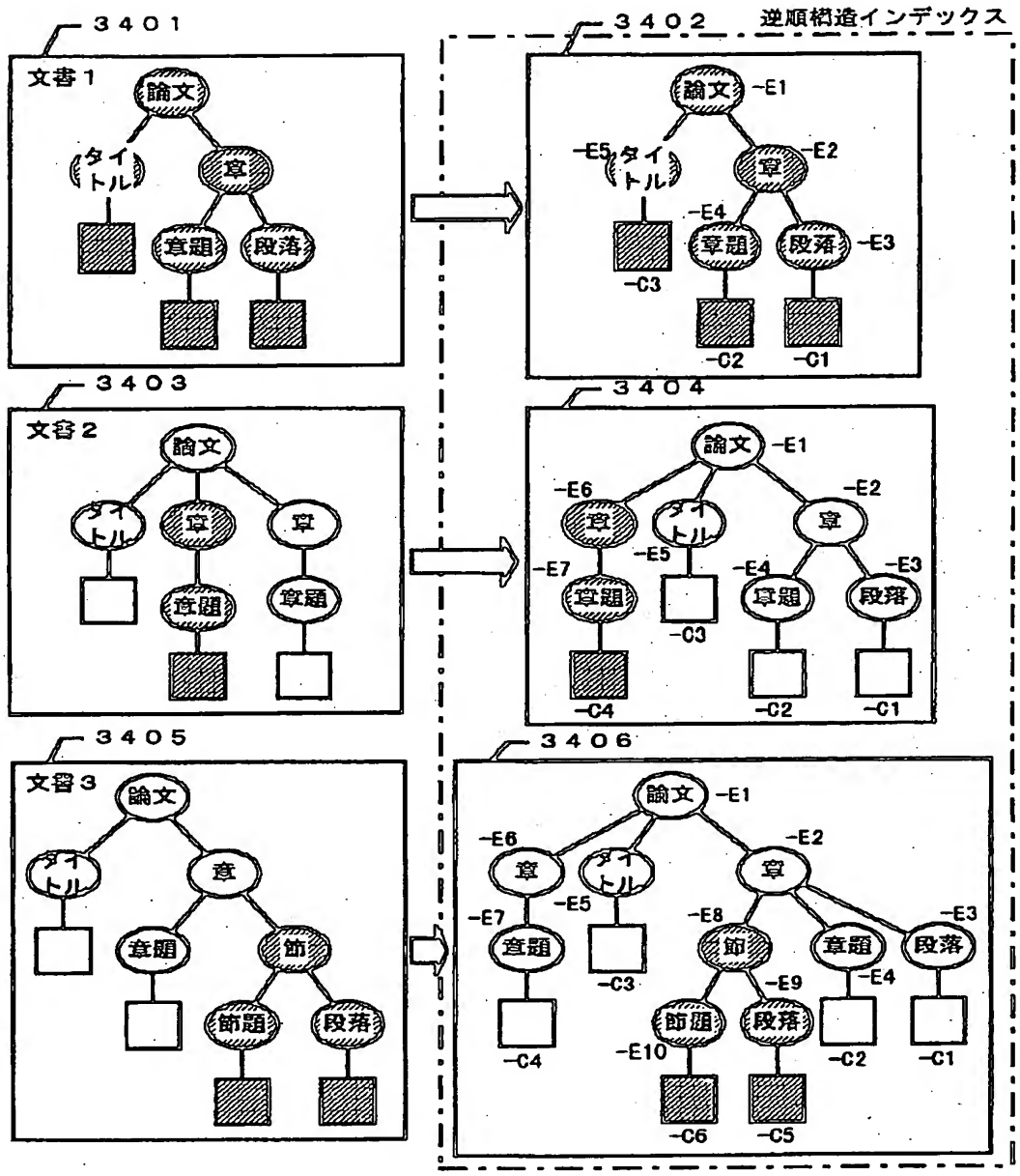
【図３１】

図３１


START
逆順構造インデックス あり？
なし
あり
3101
初期逆順構造インデックス生成
3102
逆順構造インデックス抜み込み
3103
解析済み文書データ抜み込み
3104
解析済み文書データの 木構造を構成する すべてのノードについて 繰り返し
3105
逆順構造インデックス 中に 対応メタノードあり？
3106
なし
メタノード生成、 登録
3107
逆順文脈識別子 割り当て
3108
ノードと逆順文脈識別子との 対応関係を 解析済み文書データに付加
3109
解析済み文書データ出力
3110
逆順構造インデックス出力
3111
END

【図３４】

図３４


３４０１
文書１
論文
タイトル
章
章題
段落
３４０２
逆順構造インデックス
論文 -E1
タイトル -E5
章 -E2
章題 -E4
段落 -E3
-C3
-C2
-C1
３４０３
文容２
論文
タイトル
章
章
章題
章題
３４０４
論文 -E1
章 -E6
タイトル
章 -E2
章題 -E7
-E5
章題 -E4
段落 -E3
-C3
-C4
-C2
-C1
３４０５
文容３
論文
タイトル
章
章題
節
節題
段落
３４０６
論文 -E1
章 -E6
タイトル
章 -E2
章題 -E7
-E5
節 -E8
章題 -E3
段落
-C3
-E9
-E4
節題
段落
-E10
-C4
-C2
-C1
-C6
-C5

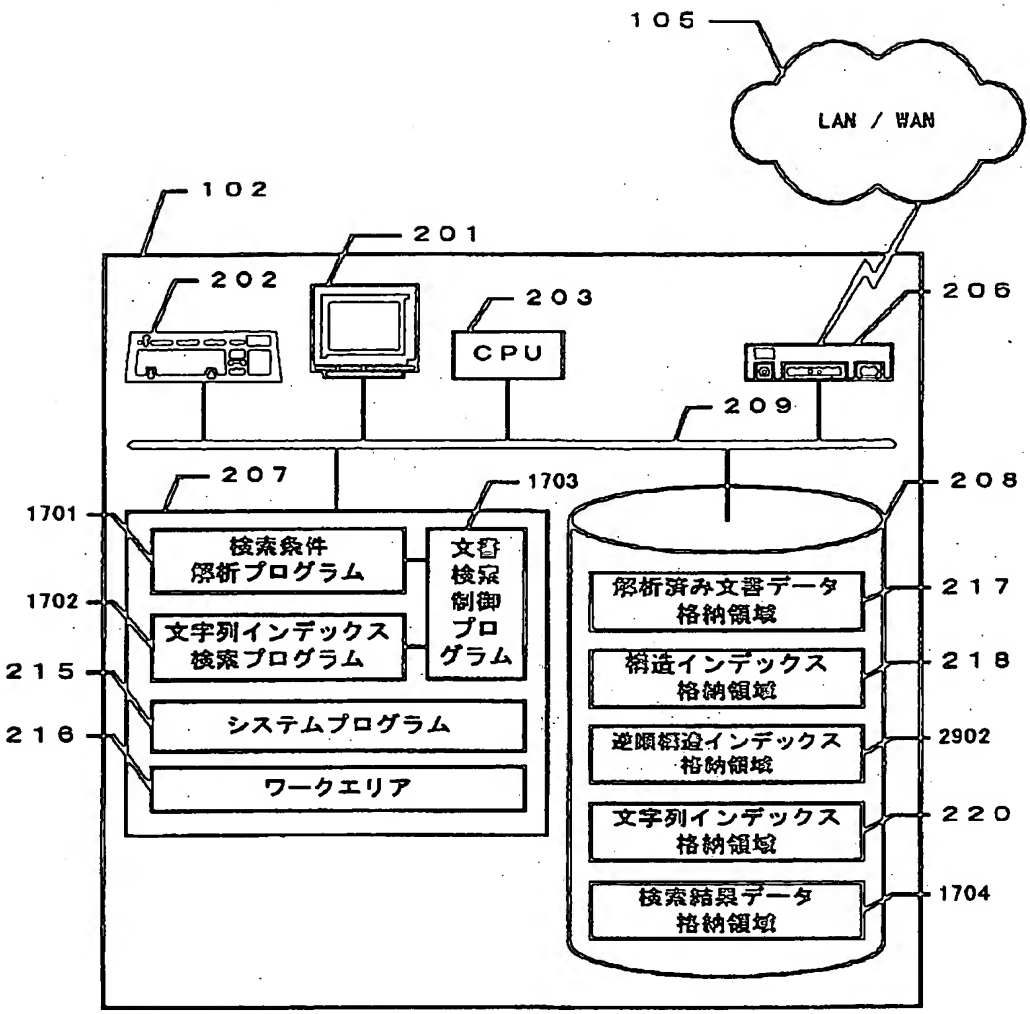
【図３６】

図３６

文書識別子	文脈識別子	逆順文脈識別子	内容文字列
D1	C1	-C351	“SGML文書変換言語の開発とその適用事例”
	C2	-C350	“高橋亨”
	C3	-C349	“東野純一”
	C4	-C348	“星幸雄”
	C5	-C347	“1996年10月23日”
	C6	-C346	“はじめに”
	C7	-C345	“文書記述にSGMLを用いることによって…”
	C8	-C344	“作成したSGML文書をさまざまな…”
	⋮	⋮	⋮
	C15	-C337	“適用事例”
	C16	-C336	“背景”
	C17	-C335	“現在、ISOでは…”
	⋮	⋮	⋮
	C129	-C223	“変換処理の実例”
	C130	-C222	“数式の変換”
	C131	-C221	“JIS規格DTDでは、基本的には数式を…”
	C132	-C220	“ただし、行列式の場合には…”
	⋮	⋮	⋮

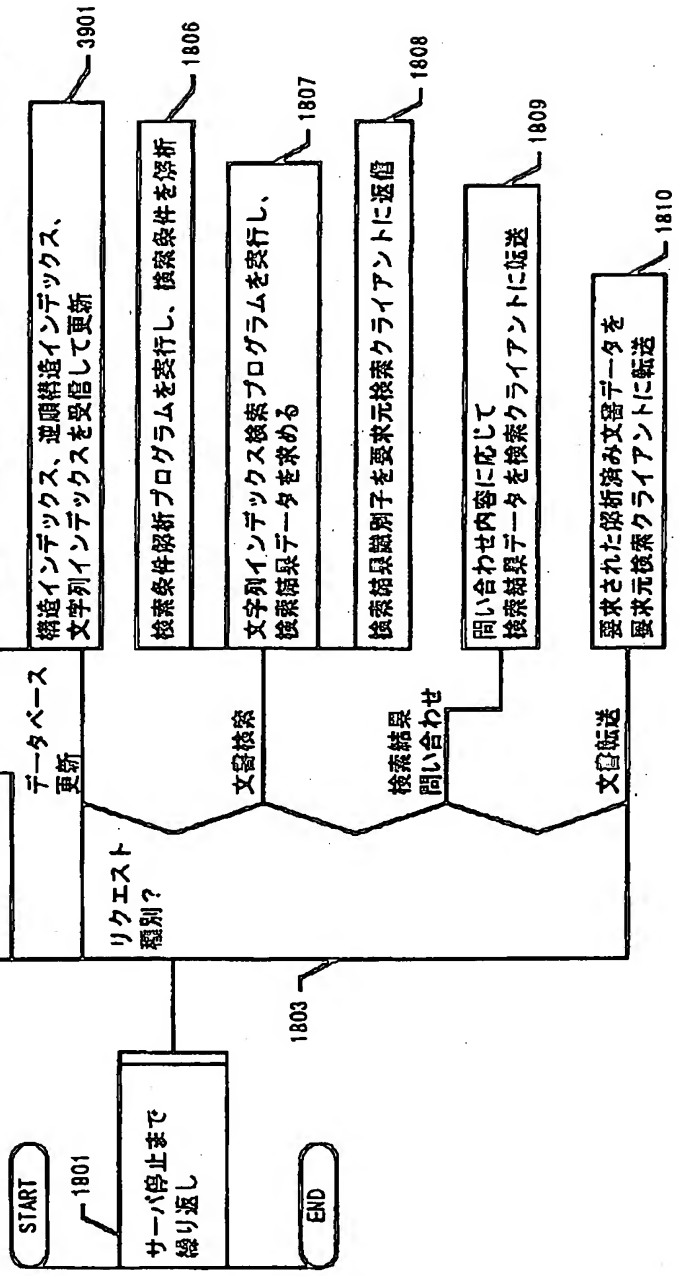
【図３８】

図３８


１０５
LAN / WAN
１０２
２０１
２０２
２０３
CPU
２０６
２０９
２０７
１７０３
２０８
１７０１
検索条件
解析プログラム
文書
検索
制御
プロ
グラム
１７０２
文字列インデックス
検索プログラム
２１５
システムプログラム
２１６
ワークエリア
解析済み文書データ
格納領域
２１７
構造インデックス
格納領域
２１８
逆順構造インデックス
格納領域
２９０２
文字列インデックス
格納領域
２２０
検索結果データ
格納領域
１７０４

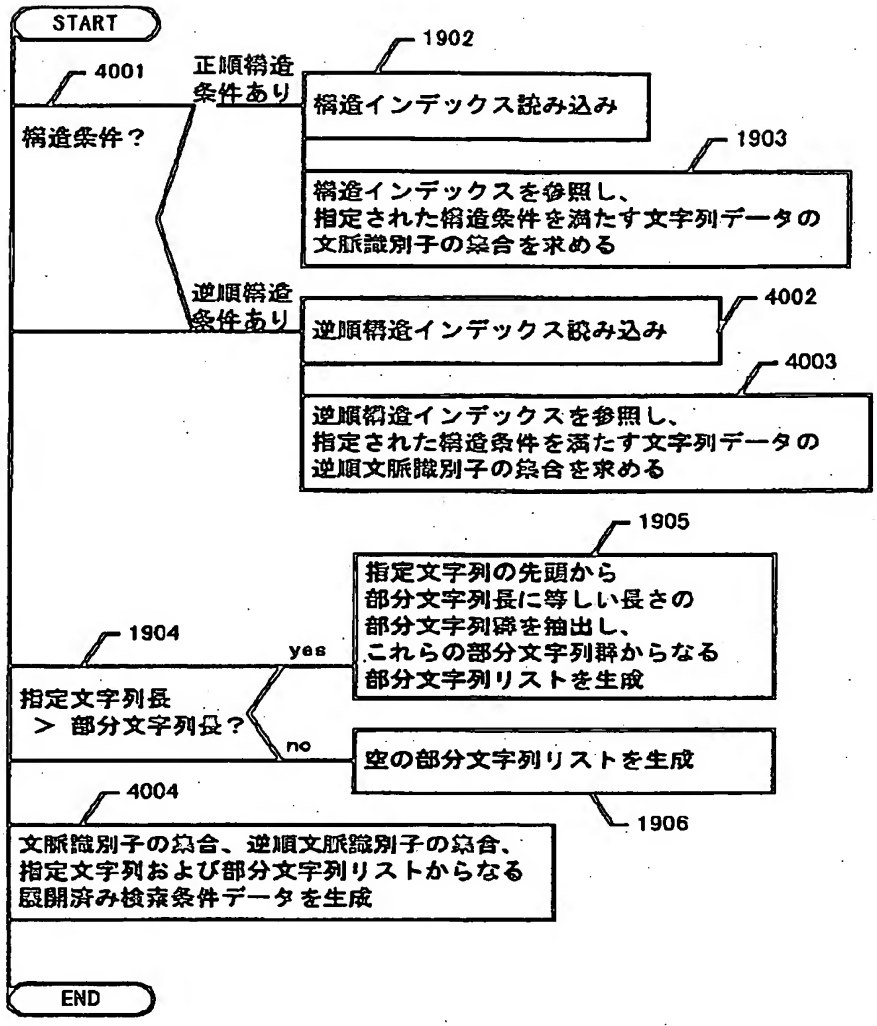
【図３９】

図３９


START
1801
サーバ停止まで
繰り返し
END
1803
1802
リクエスト受信
リクエスト
種別？
データベース
更新
文書検索
検索結果
問い合わせ
文書転送
1804
新規登録文書の解析済み文書データを受信して追加登録
3901
構造インデックス、逆順構造インデックス、
文字列インデックスを受信して更新
1806
検索条件解析プログラムを実行し、検索条件を解析
1807
文字列インデックス検索プログラムを実行し、
検索結果データを求める
1808
検索結果識別子を要求元検索クライアントに返信
1809
問い合わせ内容に応じて
検索結果データを検索クライアントに転送
1810
要求された解析済み文書データを
要求元検索クライアントに転送

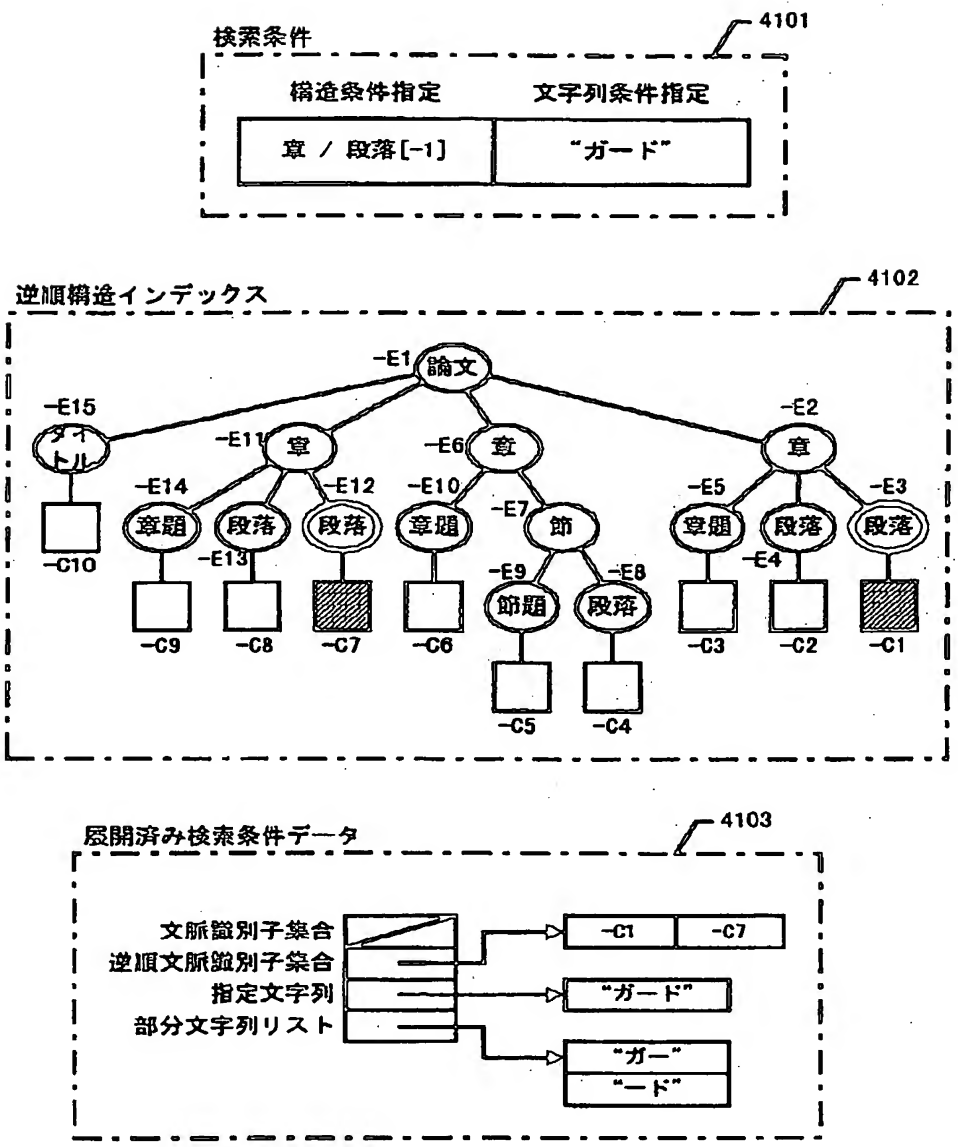
【図４０】

図４０


START
4001
構造条件？
正順構造
条件あり
1902
構造インデックス読み込み
1903
構造インデックスを参照し、
指定された構造条件を満たす文字列データの
文脈識別子の集合を求める
逆順構造
条件あり
4002
逆順構造インデックス読み込み
4003
逆順構造インデックスを参照し、
指定された構造条件を満たす文字列データの
逆順文脈識別子の集合を求める
1905
指定文字列の先頭から
部分文字列長に等しい長さの
部分文字列群を抽出し、
これらの部分文字列群からなる
部分文字列リストを生成
1904
yes
指定文字列長
＞ 部分文字列長？
no
空の部分文字列リストを生成
1906
4004
文脈識別子の集合、逆順文脈識別子の集合、
指定文字列および部分文字列リストからなる
展開済み検索条件データを生成
END

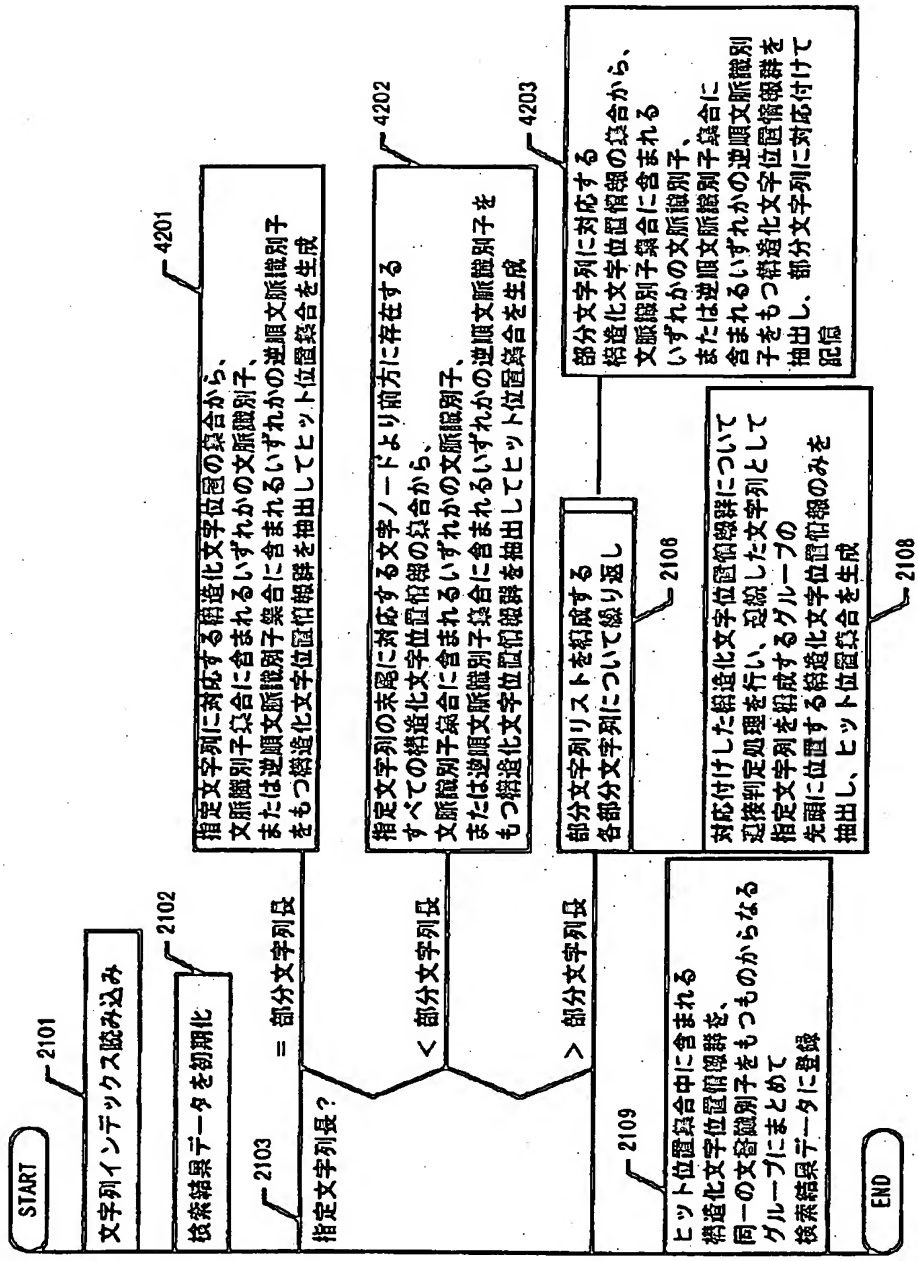
【図４１】

図４１


検索条件
4101
構造条件指定
文字列条件指定
章 / 段落[-1]
"ガード"
逆順構造インデックス
4102
-E1 論文
-E15 タイトル
-E11 章
-E6 章
-E2 章
-E14 章題
-E12 段落
-E13
-E10 章題
-E7 節
-E5 章題
-E4
-E3 段落
-E9 節題
-E8 段落
-C10
-C9
-C8
-C7
-C6
-C5
-C4
-C3
-C2
-C1
展開済み検索条件データ
4103
文脈識別子集合
逆順文脈識別子集合
指定文字列
部分文字列リスト
-C1
-C7
"ガード"
"ガー"
"ード"

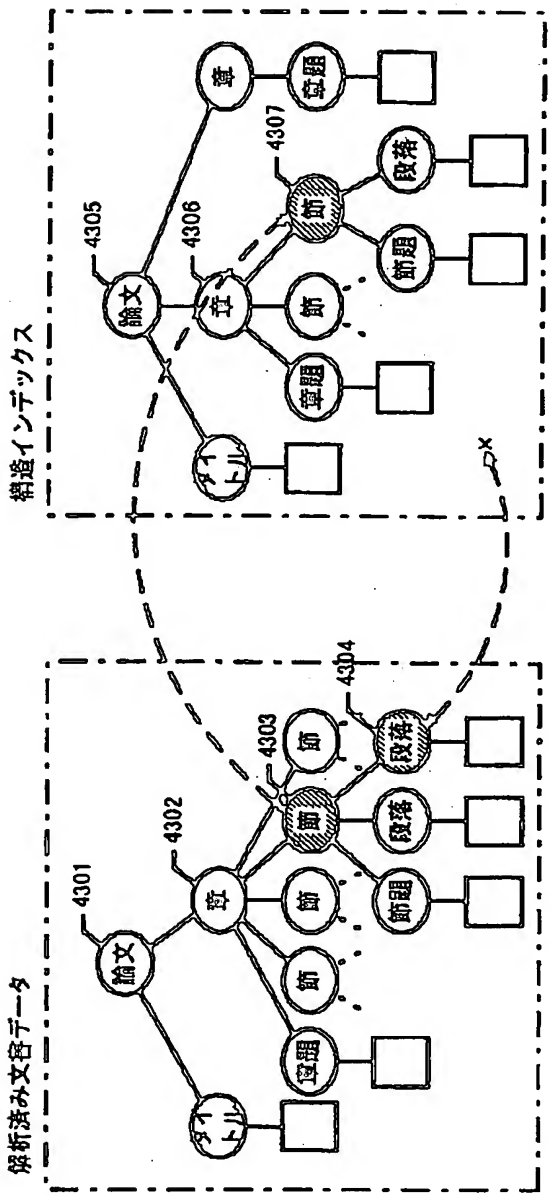
【図４２】

図４２


START
2101
文字列インデックス読み込み
2102
検索結果データを初期化
2103
指定文字列長？
= 部分文字列長
指定文字列に対応する構造化文字位置の袋合から、
文脈識別子袋合に含まれるいずれかの文脈識別子、
または逆順文脈識別子集合に含まれるいずれかの逆順文脈識別子
をもつ構造化文字位置情報群を抽出してヒット位置袋合を生成
4201
< 部分文字列長
指定文字列の末尾に対応する文字ノードより前方に存在する
すべての構造化文字位置情報の袋合から、
文脈識別子袋合に含まれるいずれかの文脈識別子、
または逆順文脈識別子袋合に含まれるいずれかの逆順文脈識別子を
もつ構造化文字位置情報群を抽出してヒット位置袋合を生成
4202
> 部分文字列長
部分文字列リストを初成する
各部分文字列について繰り返し
2106
部分文字列に対応する
構造化文字位置情報の袋合から、
文脈識別子袋合に含まれる
いずれかの文脈識別子、
または逆順文脈識別子袋合に
含まれるいずれかの逆順文脈識別
子をもつ構造化文字位置情報群を
抽出し、部分文字列に対応付けて
記憶
4203
2109
ヒット位置袋合中に含まれる
構造化文字位置情報群を、
同一の文容識別子をもつものからなる
グループにまとめて
検索結果データに登録
対応付けした構造化文字位置情報群について
辺接判定処理を行い、辺接した文字列として
指定文字列を初成するグループの
先頭に位置する構造化文字位置情報のみを
抽出し、ヒット位置袋合を生成
2108
END

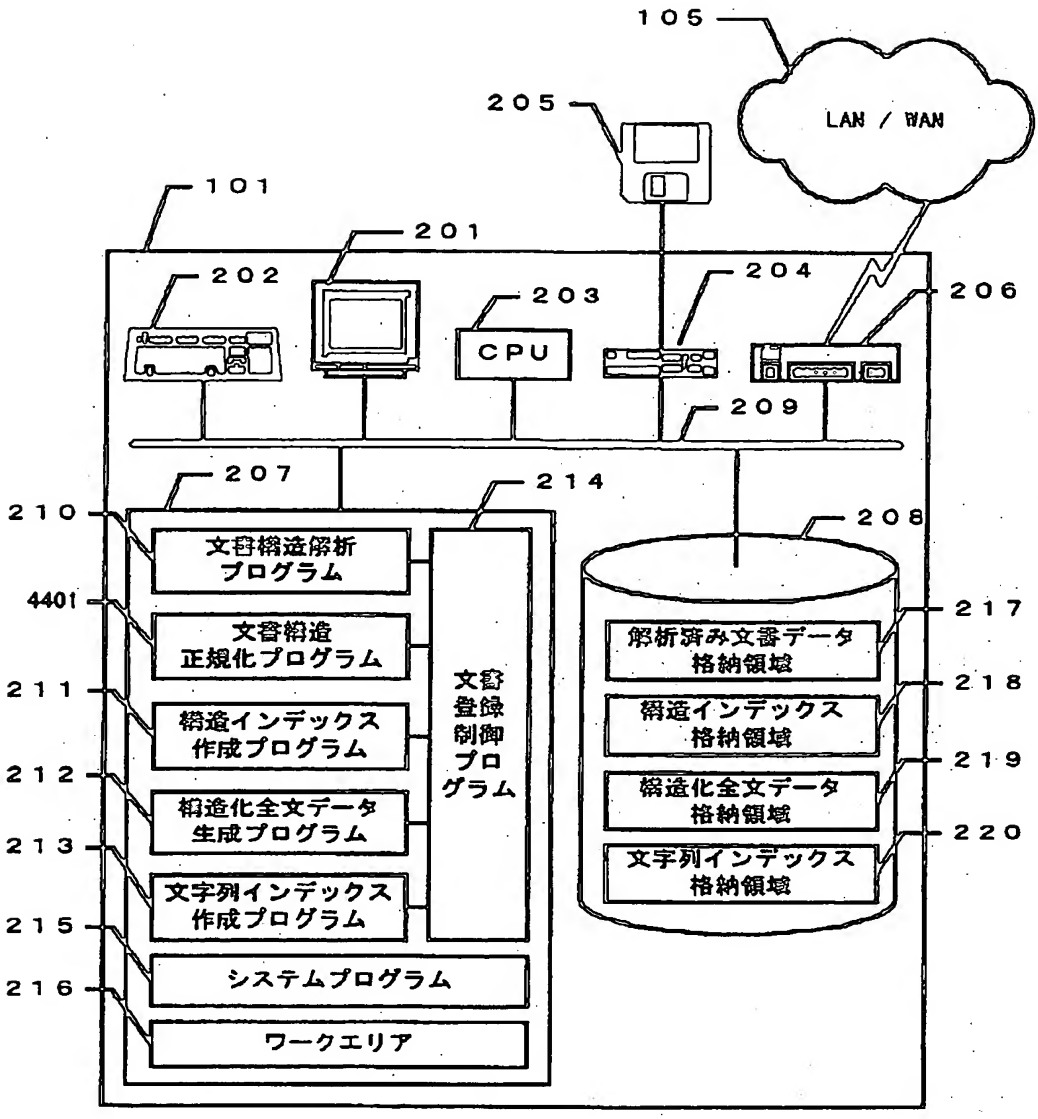
【図４３】

図４３


構造インデックス
解析済み文書データ
論文
章
節
段落
節題
章題
4301
4302
4303
4304
4305
4306
4307

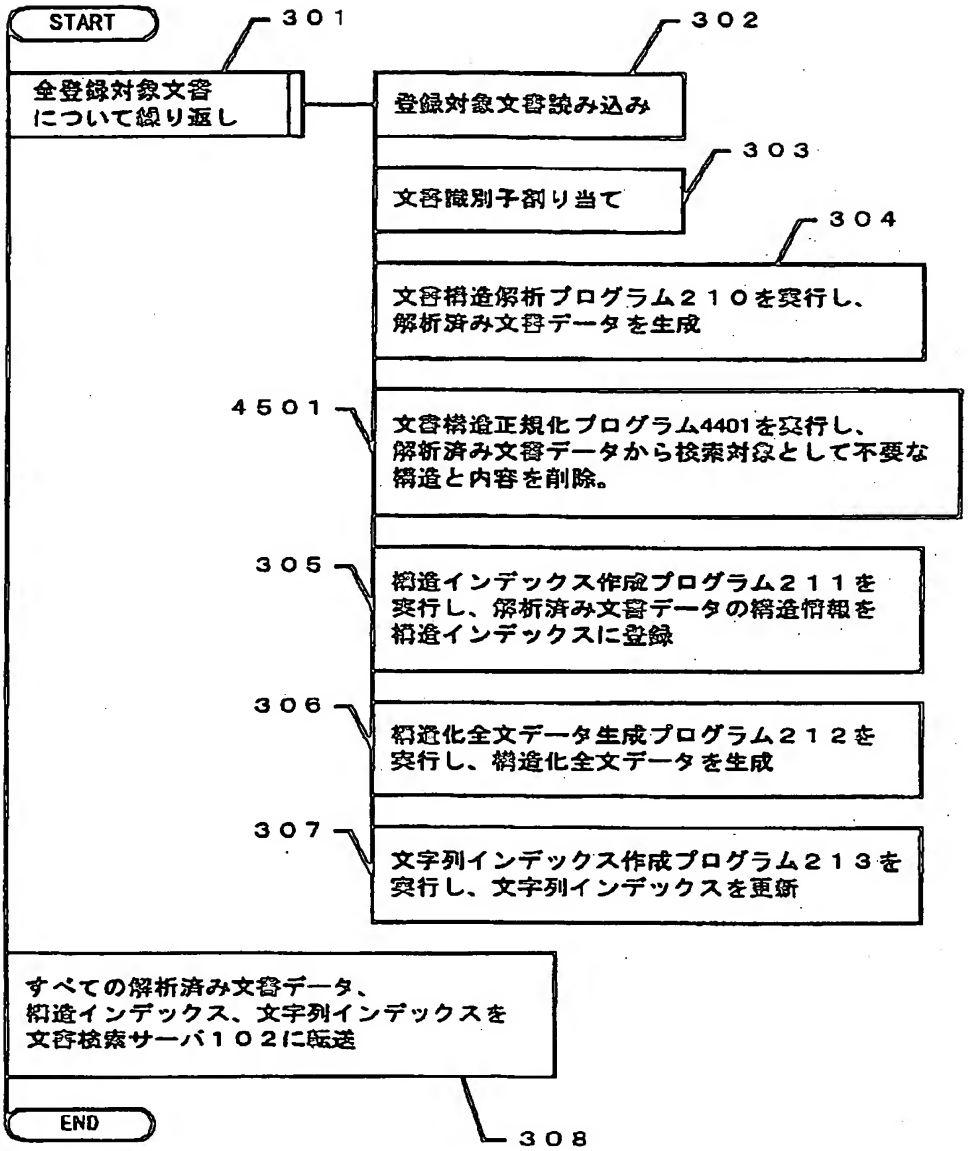
【図４４】

図４４


105
LAN / WAN
205
101
201
202
203
CPU
204
206
209
207
214
208
210
文書構造解析プログラム
4401
文書構造正規化プログラム
211
構造インデックス作成プログラム
212
構造化全文データ生成プログラム
213
文字列インデックス作成プログラム
215
システムプログラム
216
ワークエリア
文書登録制御プログラム
解析済み文書データ格納領域
217
構造インデックス格納領域
218
構造化全文データ格納領域
219
文字列インデックス格納領域
220

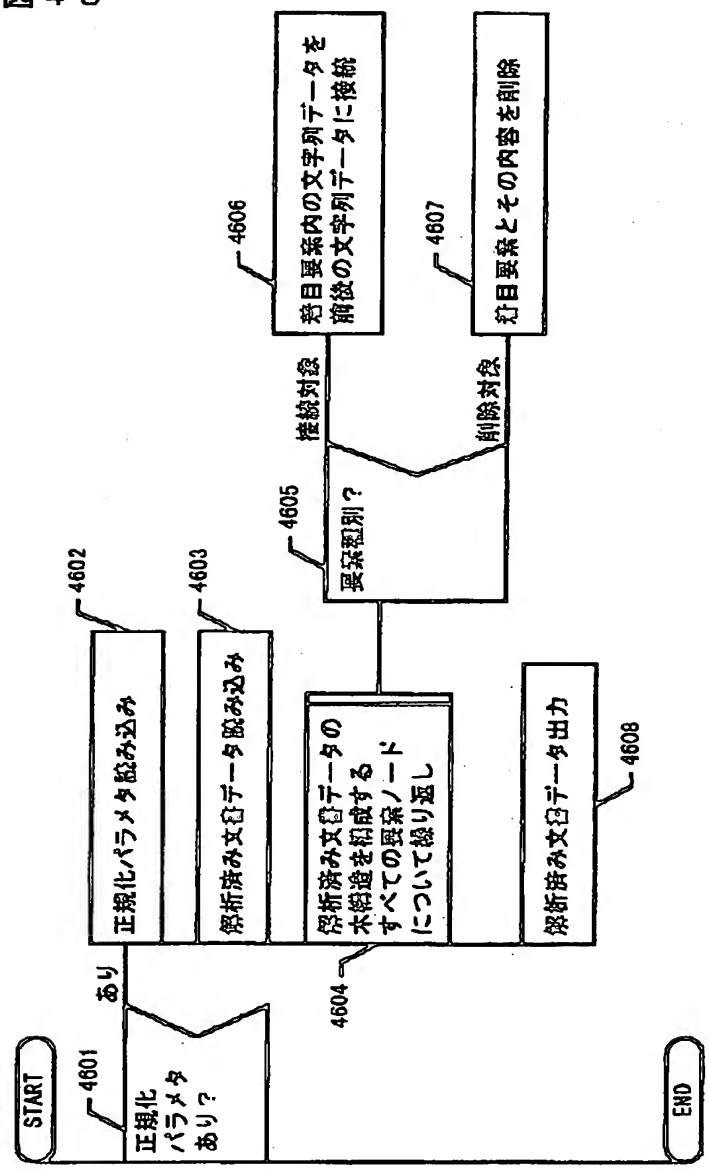
【図４５】

図４５


START
３０１
全登録対象文書
について繰り返し
３０２
登録対象文書読み込み
３０３
文書識別子割り当て
３０４
文書構造解析プログラム２１０を実行し、
解析済み文書データを生成
４５０１
文書構造正規化プログラム4401を実行し、
解析済み文書データから検索対象として不要な
構造と内容を削除。
３０５
構造インデックス作成プログラム２１１を
実行し、解析済み文書データの構造情報を
構造インデックスに登録
３０６
構造化全文データ生成プログラム２１２を
実行し、構造化全文データを生成
３０７
文字列インデックス作成プログラム２１３を
実行し、文字列インデックスを更新
すべての解析済み文書データ、
構造インデックス、文字列インデックスを
文書検索サーバ１０２に転送
３０８
END

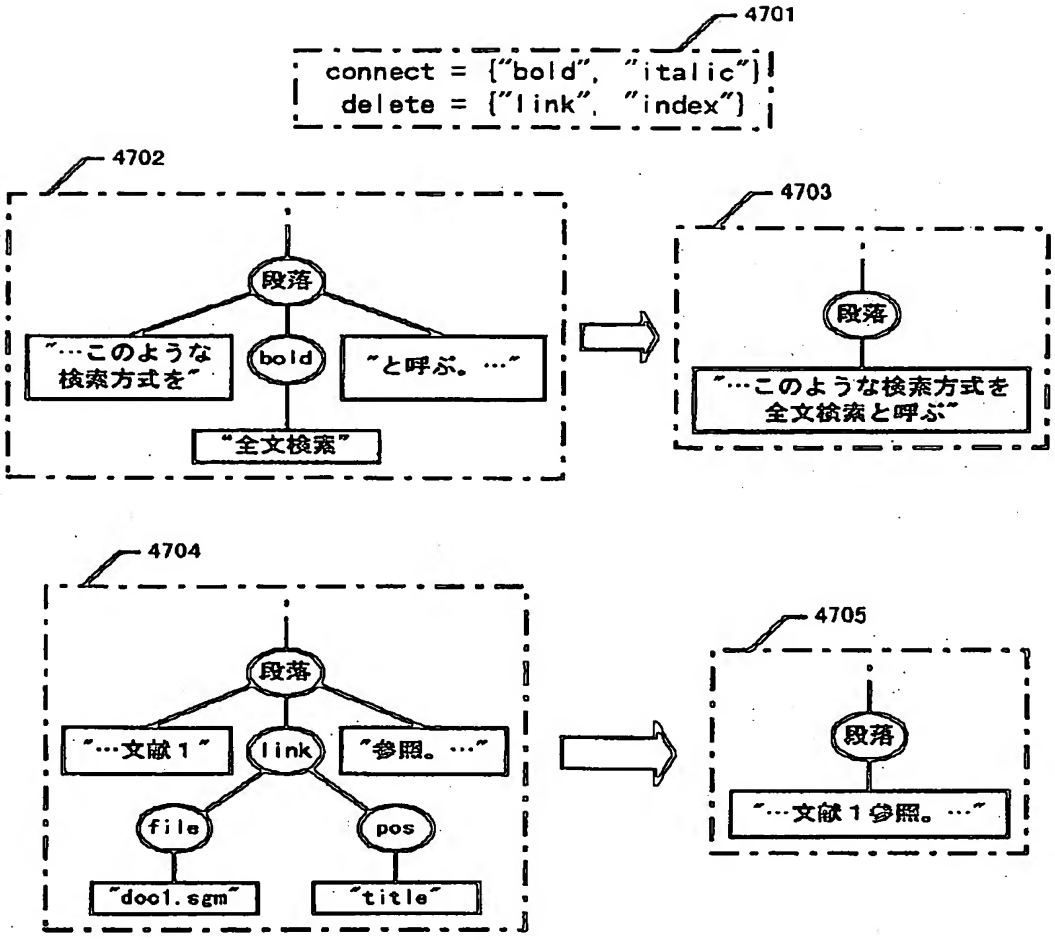
【図４６】

図４６


START
4601
正規化パラメタあり？
あり
4602
正規化パラメタ読み込み
4603
解析済み文台データ読み込み
4604
解析済み文台データの木構造を構成するすべての要素ノードについて繰り返し
4605
要素種別？
接続対象
4606
対目要素内の文字列データを前後の文字列データに接続
削除対象
4607
灯目要素とその内容を削除
4608
解析済み文台データ出力
END

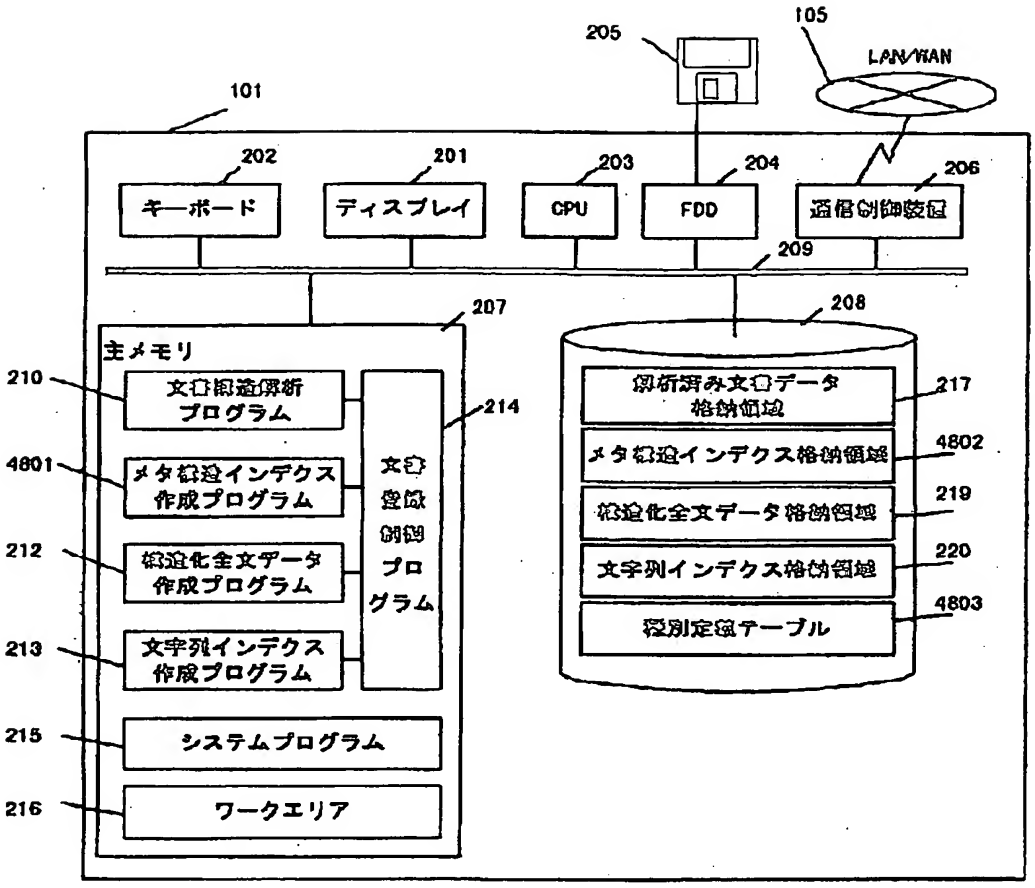
【図４７】

図４７


4701
connect = {"bold", "italic"}
delete = {"link", "index"}
4702
段落
"…このような検索方式を"
bold
"と呼ぶ。…"
"全文検索"
4703
段落
"…このような検索方式を全文検索と呼ぶ"
4704
段落
"…文献１"
link
"参照。…"
file
pos
"doc1.sgm"
"title"
4705
段落
"…文献１参照。…"

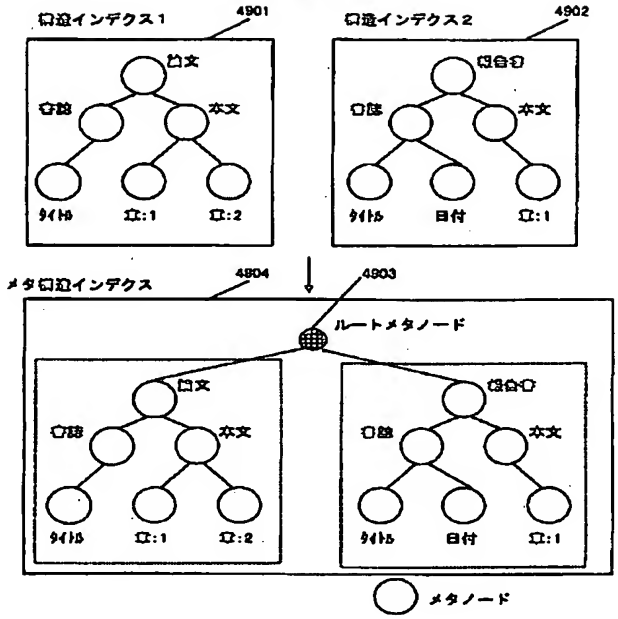
【図４８】

図４８


205
105
LAN/WAN
101
202
201
203
204
206
キーボード
ディスプレイ
CPU
FDD
通信制御装置
209
207
208
主メモリ
210
文書構造解析プログラム
214
4801
メタ構造インデクス作成プログラム
文書管理制御プログラム
212
構造化全文データ作成プログラム
213
文字列インデクス作成プログラム
215
システムプログラム
216
ワークエリア
解析済み文書データ格納領域
217
メタ構造インデクス格納領域
4802
構造化全文データ格納領域
219
文字列インデクス格納領域
220
識別定義テーブル
4803

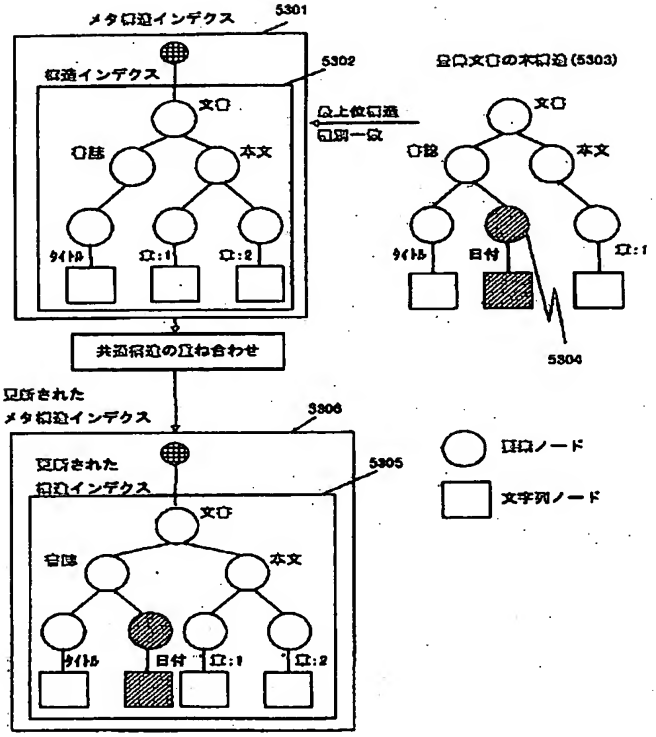
【図４９】

図４９



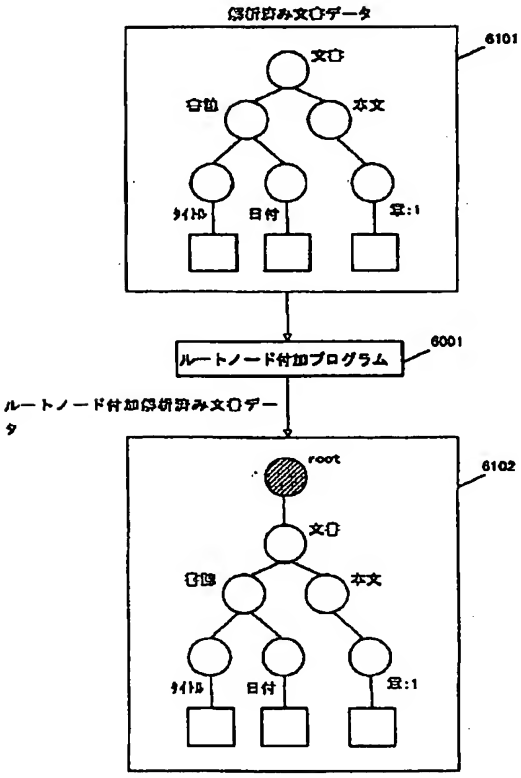
【図５３】

図５３



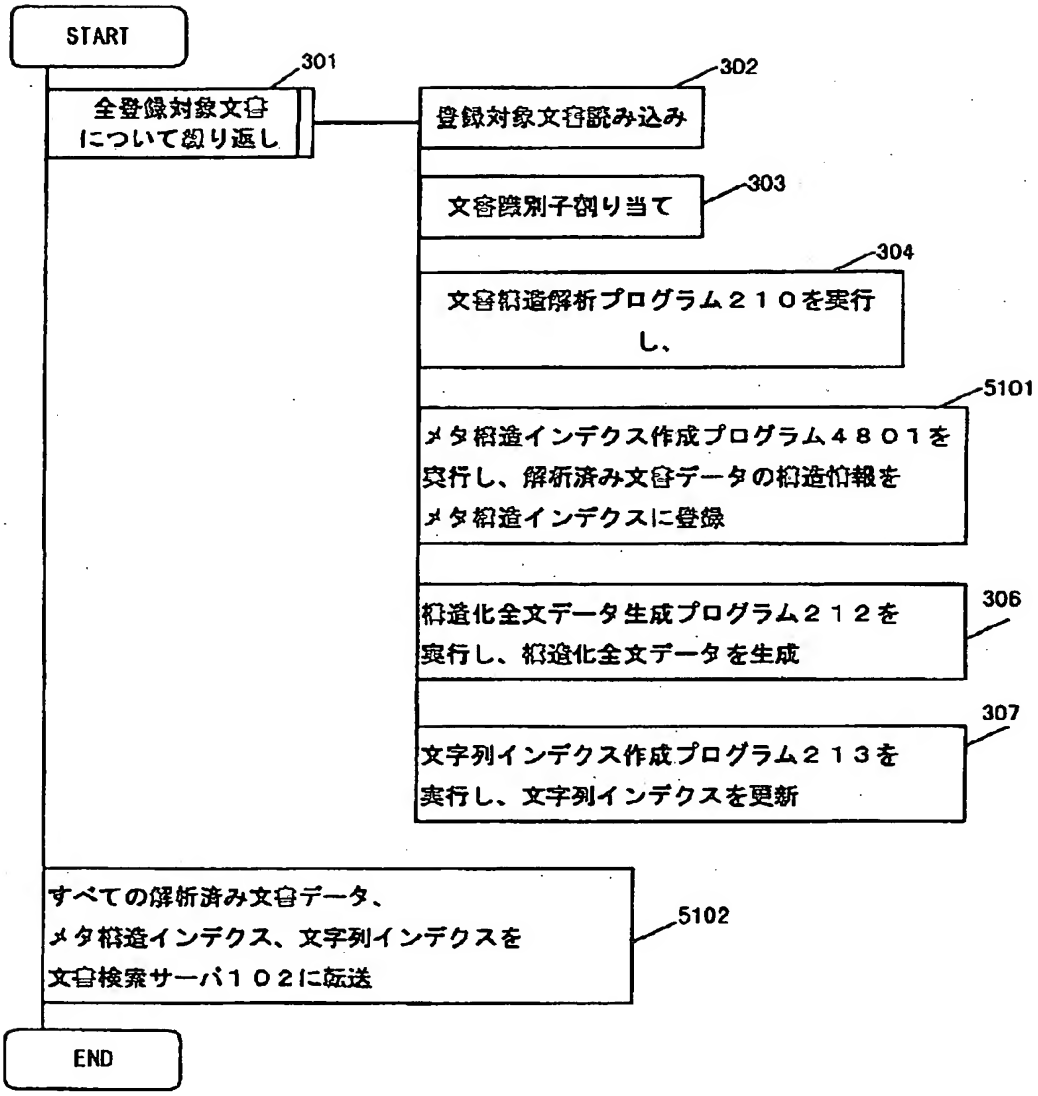
【図６１】

図６１



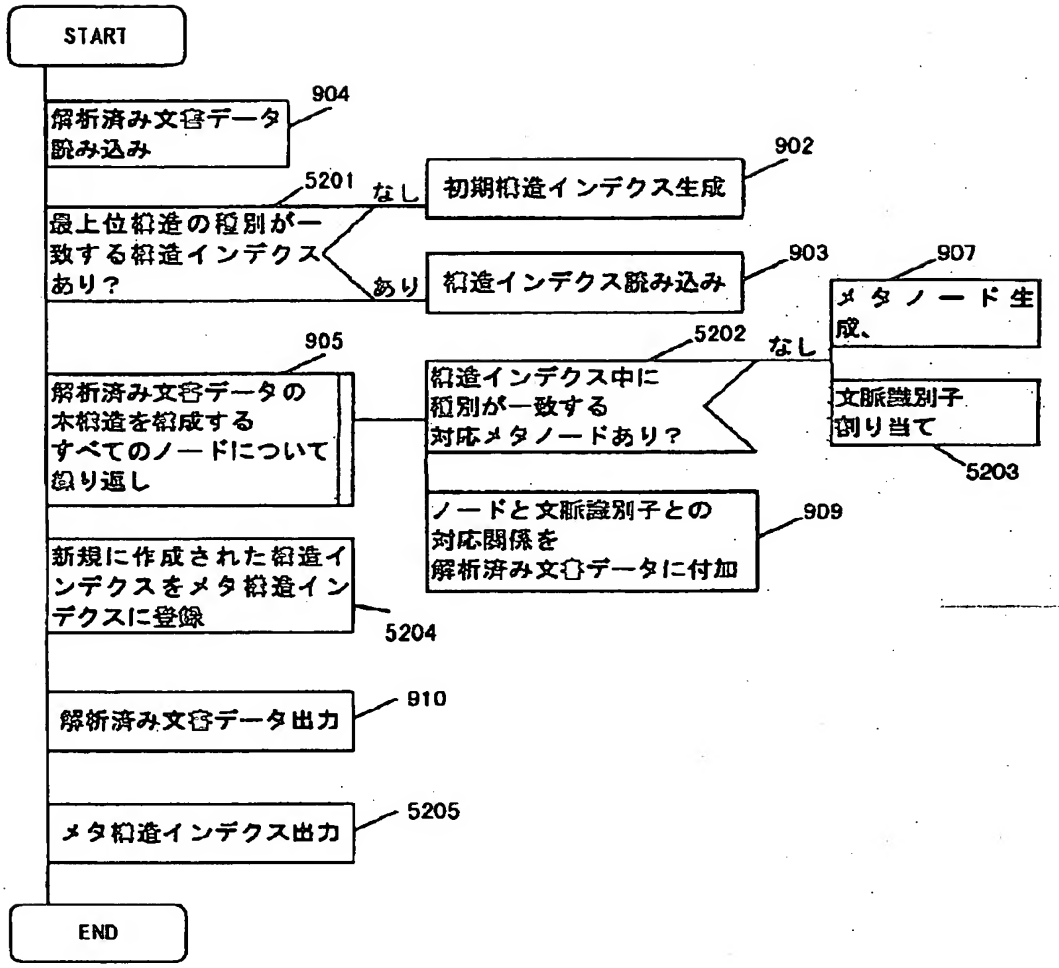
【図５１】

図５１


START
301
全登録対象文書について繰り返し
302
登録対象文書読み込み
303
文書識別子割り当て
304
文書構造解析プログラム２１０を実行し、
5101
メタ構造インデクス作成プログラム４８０１を実行し、解析済み文書データの構造情報をメタ構造インデクスに登録
306
構造化全文データ生成プログラム２１２を実行し、構造化全文データを生成
307
文字列インデクス作成プログラム２１３を実行し、文字列インデクスを更新
すべての解析済み文書データ、メタ構造インデクス、文字列インデクスを文書検索サーバ１０２に転送
5102
END

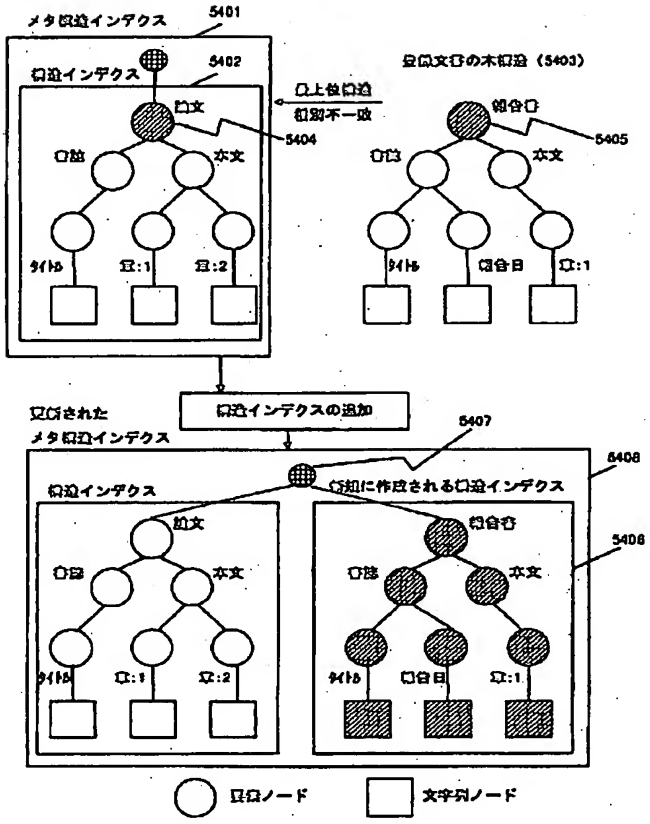
【図５２】

図５２


START
解析済み文書データ読み込み
904
最上位構造の種別が一致する構造インデクスあり？
5201
なし
あり
初期構造インデクス生成
902
構造インデクス読み込み
903
構造インデクス中に種別が一致する対応メタノードあり？
5202
なし
メタノード生成、
907
文脈識別子割り当て
5203
ノードと文脈識別子との対応関係を解析済み文書データに付加
909
解析済み文書データの本構造を構成するすべてのノードについて繰り返し
905
新規に作成された構造インデクスをメタ構造インデクスに登録
5204
解析済み文書データ出力
910
メタ構造インデクス出力
5205
END

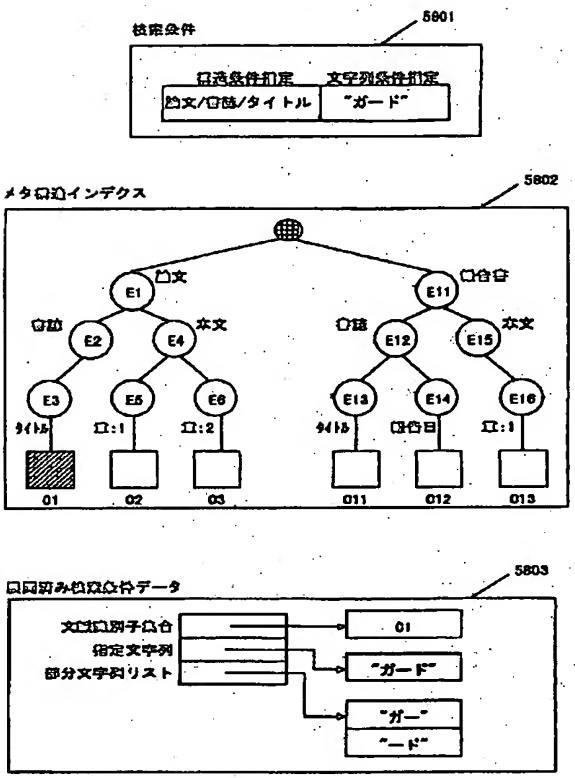
【図５４】

図５４



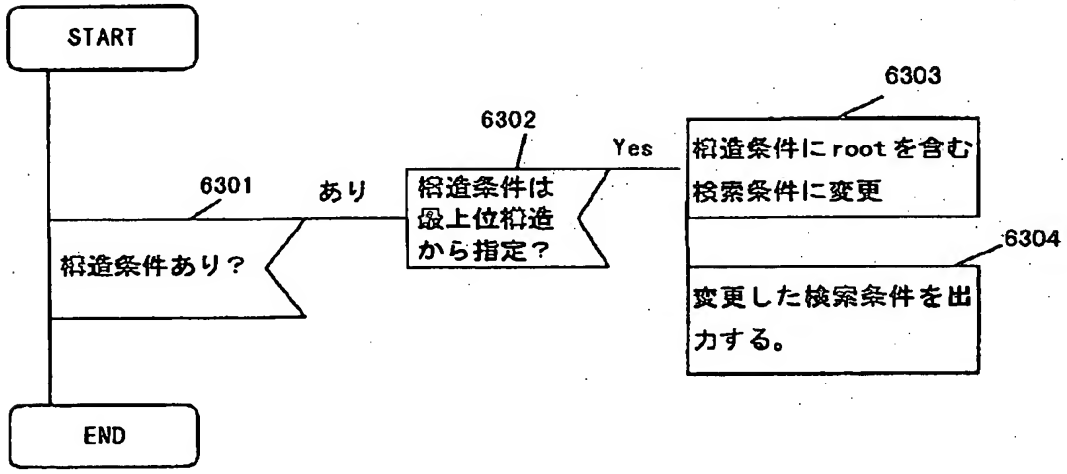
【図５８】

図５８



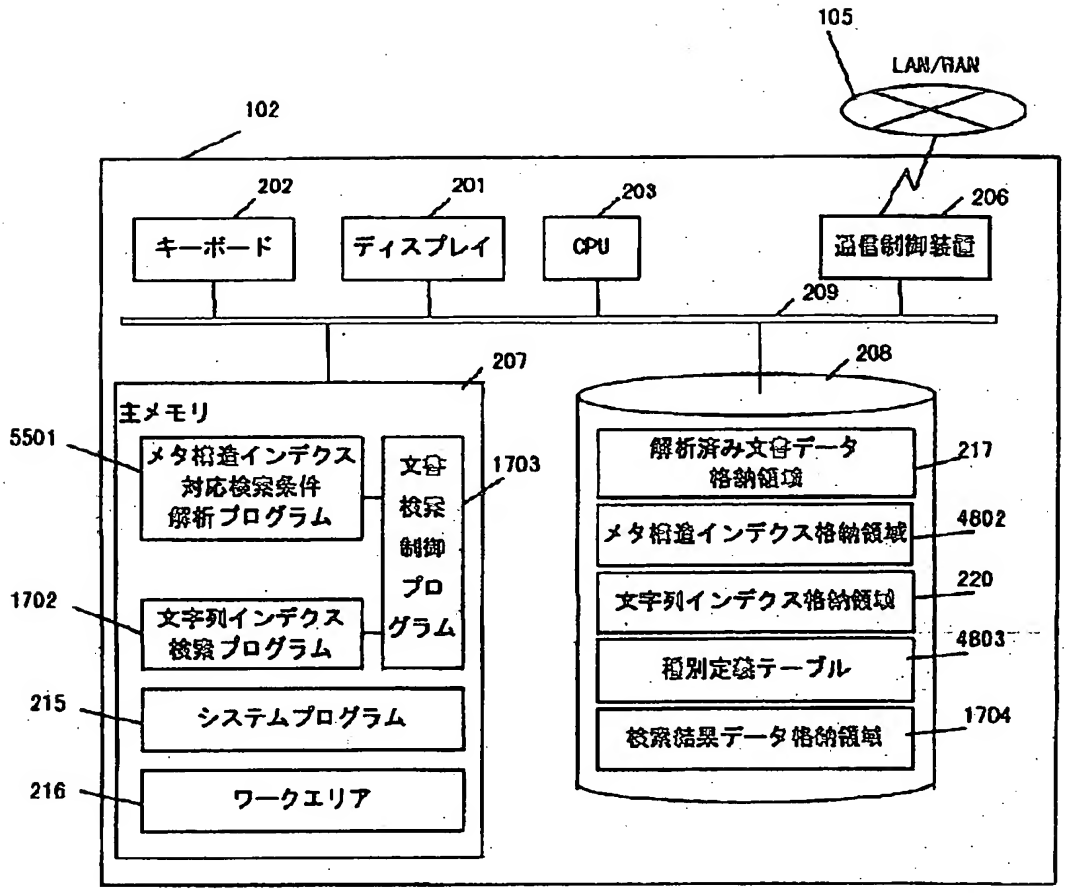
【図６３】

図６３



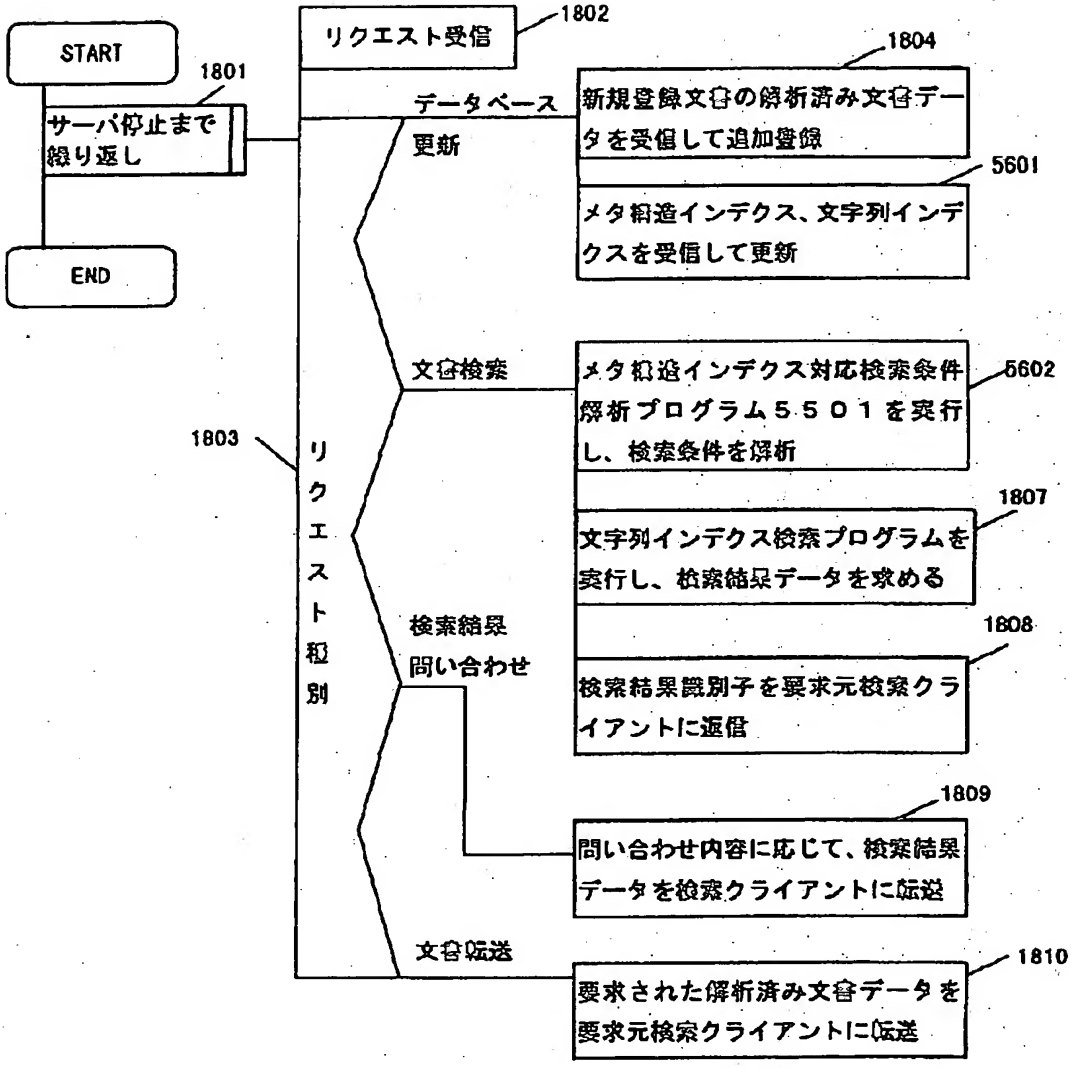
【図５５】

図５５


105
LAN/WAN
102
202
201
203
206
キーボード
ディスプレイ
CPU
通信制御装置
209
207
208
主メモリ
5501
メタ構造インデクス対応検索条件解析プログラム
文書検索制御プログラム
1703
1702
文字列インデクス検索プログラム
215
システムプログラム
216
ワークエリア
解析済み文書データ格納領域
217
メタ構造インデクス格納領域
4802
文字列インデクス格納領域
220
種別定義テーブル
4803
検索結果データ格納領域
1704

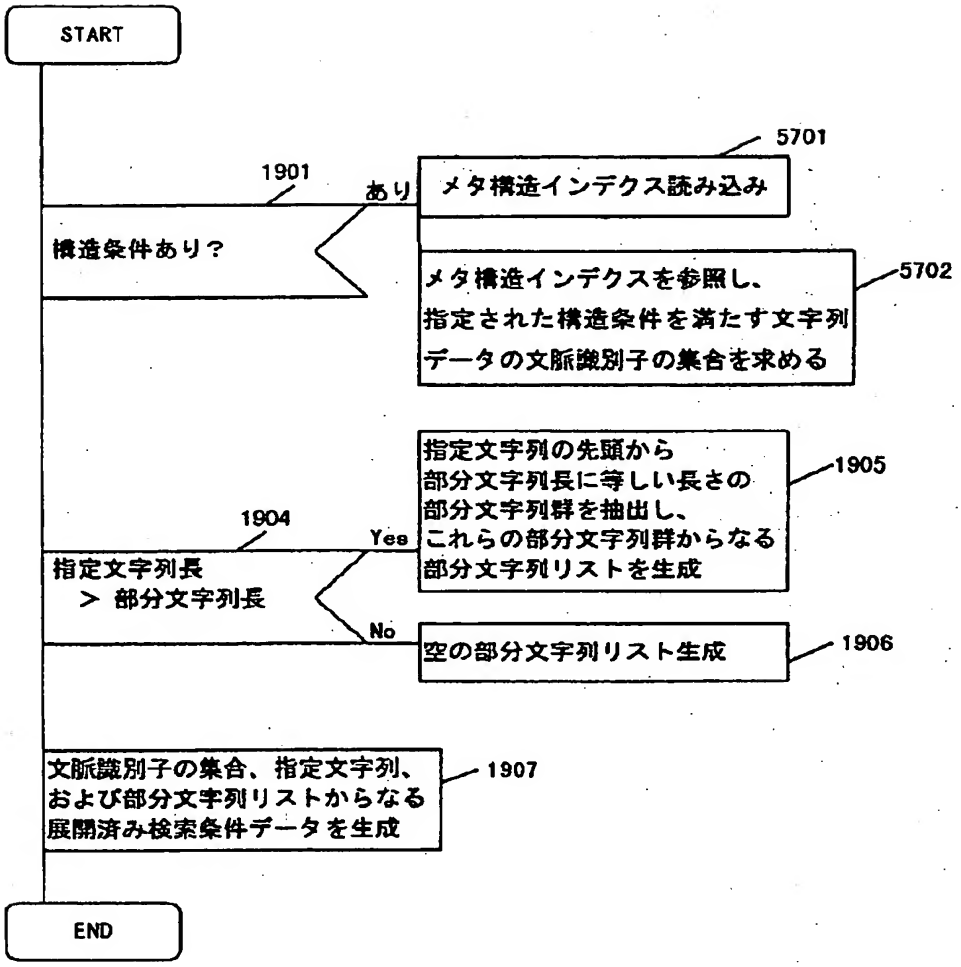
【図５６】

図５６


START
1801
サーバ停止まで繰り返し
END
リクエスト受信
1802
データベース
更新
新規登録文書の解析済み文書データを受信して追加登録
1804
メタ構造インデクス、文字列インデクスを受信して更新
5601
文書検索
メタ構造インデクス対応検索条件解析プログラム５５０１を実行し、検索条件を解析
5602
文字列インデクス検索プログラムを実行し、検索結果データを求める
1807
検索結果問い合わせ
検索結果識別子を要求元検索クライアントに返信
1808
問い合わせ内容に応じて、検索結果データを検索クライアントに転送
1809
文書転送
要求された解析済み文書データを要求元検索クライアントに転送
1810
1803
リクエスト種別

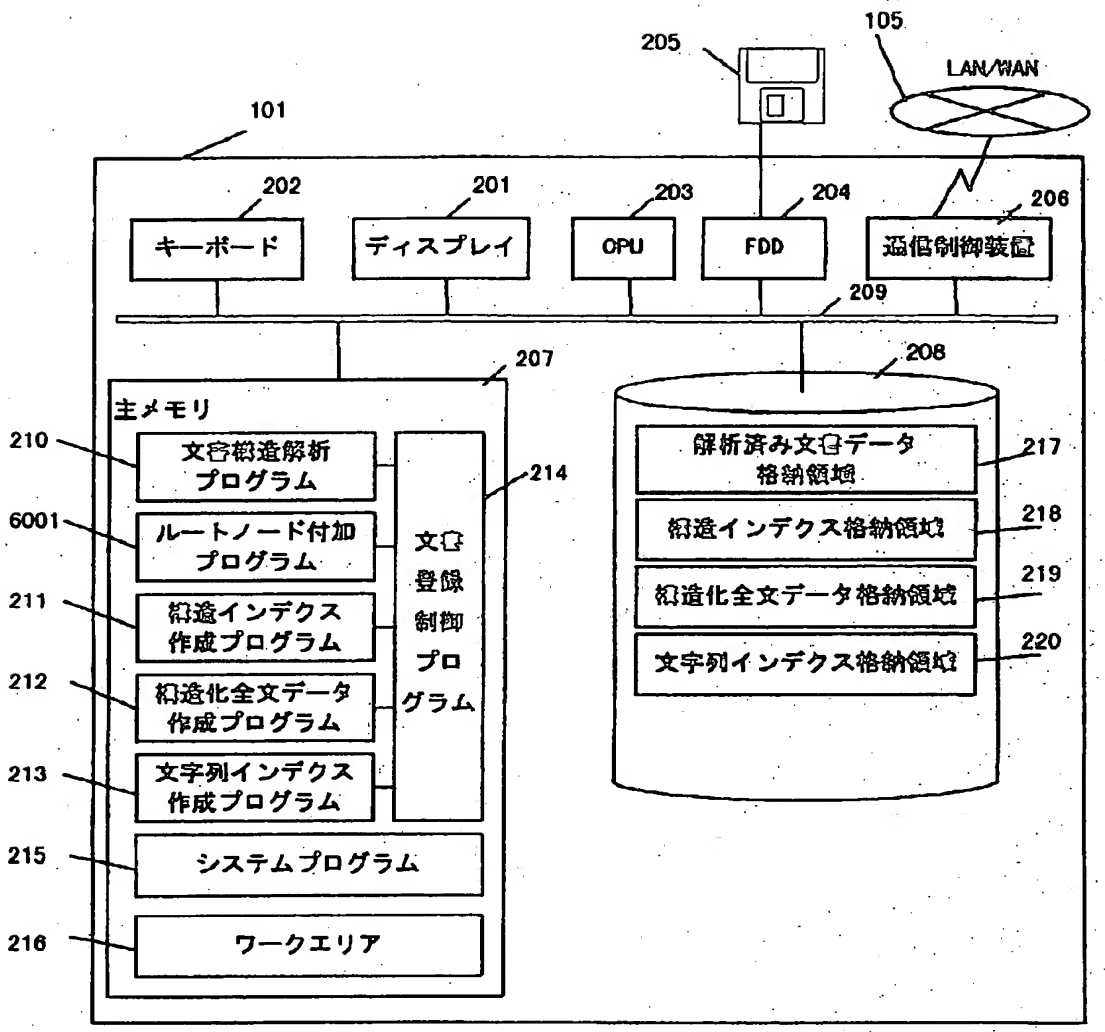
【図５７】

図５７


START
1901
あり
構造条件あり？
5701
メタ構造インデクス読み込み
メタ構造インデクスを参照し、
指定された構造条件を満たす文字列
データの文脈識別子の集合を求める
5702
1904
Yes
指定文字列長
＞ 部分文字列長
No
指定文字列の先頭から
部分文字列長に等しい長さの
部分文字列群を抽出し、
これらの部分文字列群からなる
部分文字列リストを生成
1905
空の部分文字列リスト生成
1906
文脈識別子の集合、指定文字列、
および部分文字列リストからなる
展開済み検索条件データを生成
1907
END

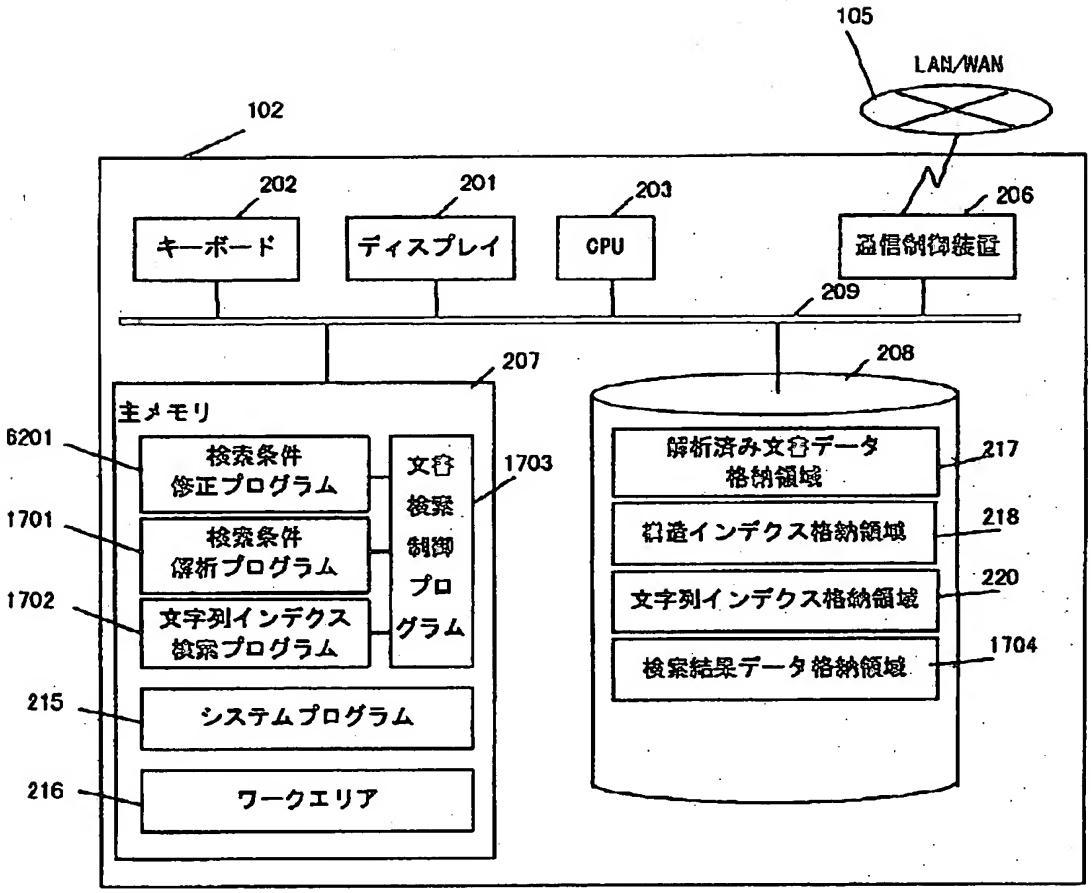
【図６０】

図６０


205
105
LAN/WAN
101
202
201
203
204
206
キーボード
ディスプレイ
CPU
FDD
通信制御装置
209
207
208
主メモリ
210
文書構造解析プログラム
6001
ルートノード付加プログラム
211
構造インデクス作成プログラム
212
構造化全文データ作成プログラム
213
文字列インデクス作成プログラム
214
文書登録制御プログラム
215
システムプログラム
216
ワークエリア
解析済み文書データ格納領域
217
構造インデクス格納領域
218
構造化全文データ格納領域
219
文字列インデクス格納領域
220

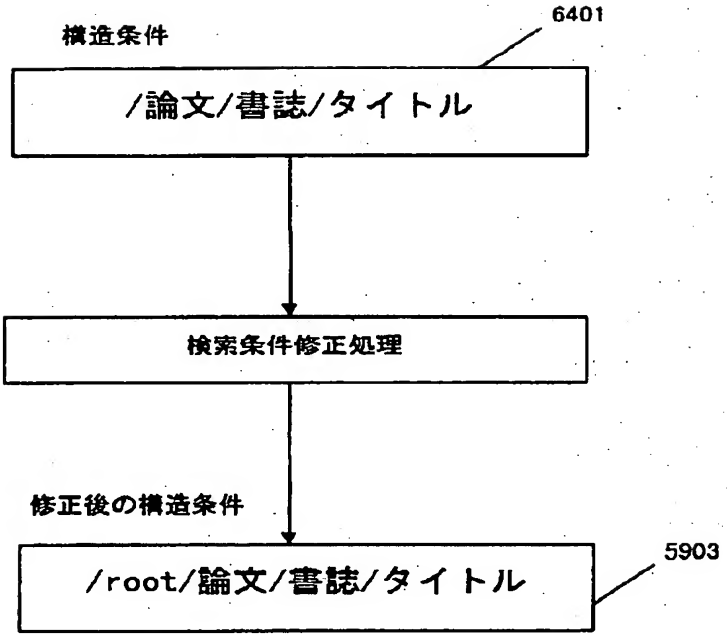
【図６２】

図６２


105
LAN/WAN
102
202
201
203
206
キーボード
ディスプレイ
CPU
通信制御装置
209
207
208
主メモリ
6201
検索条件
修正プログラム
1701
検索条件
解析プログラム
1702
文字列インデクス
検索プログラム
文書
検索
制御
プロ
グラム
1703
215
システムプログラム
216
ワークエリア
解析済み文書データ
格納領域
217
構造インデクス格納領域
218
文字列インデクス格納領域
220
検索結果データ格納領域
1704

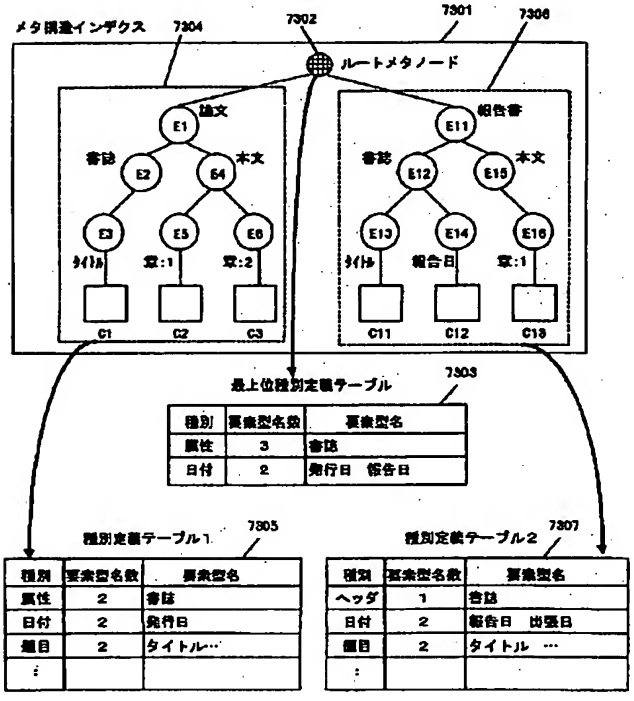
【図６４】

図６４



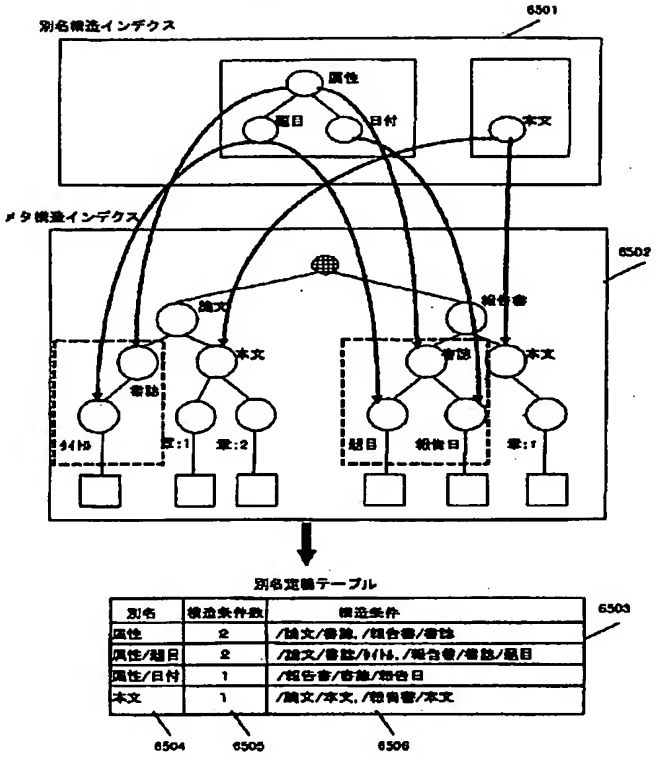
【図７３】

図７３



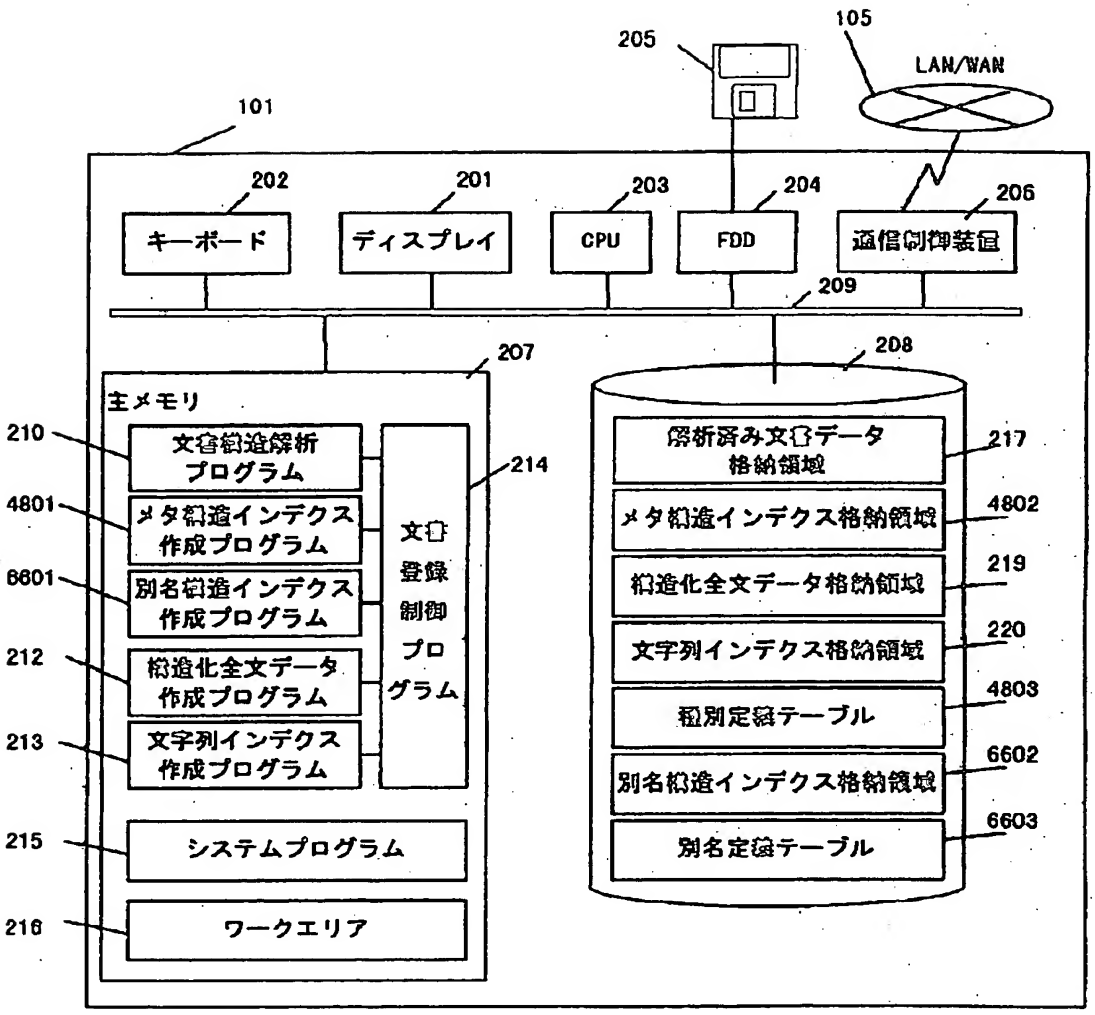
【図６５】

図６５


別名構造インデクス
6501
属性
題目
日付
本文
メタ構造インデクス
6502
論文
報告書
本文
著話
本文
著話
タイトル
章:1
章:2
題目
報告日
章:1
別名定義テーブル
6503
別名
構造条件数
構造条件
属性
2
/論文/著話, /報告書/著話
属性/題目
2
/論文/著話/タイトル, /報告書/著話/題目
属性/日付
1
/報告書/著話/報告日
本文
1
/論文/本文, /報告書/本文
6504
6505
6506

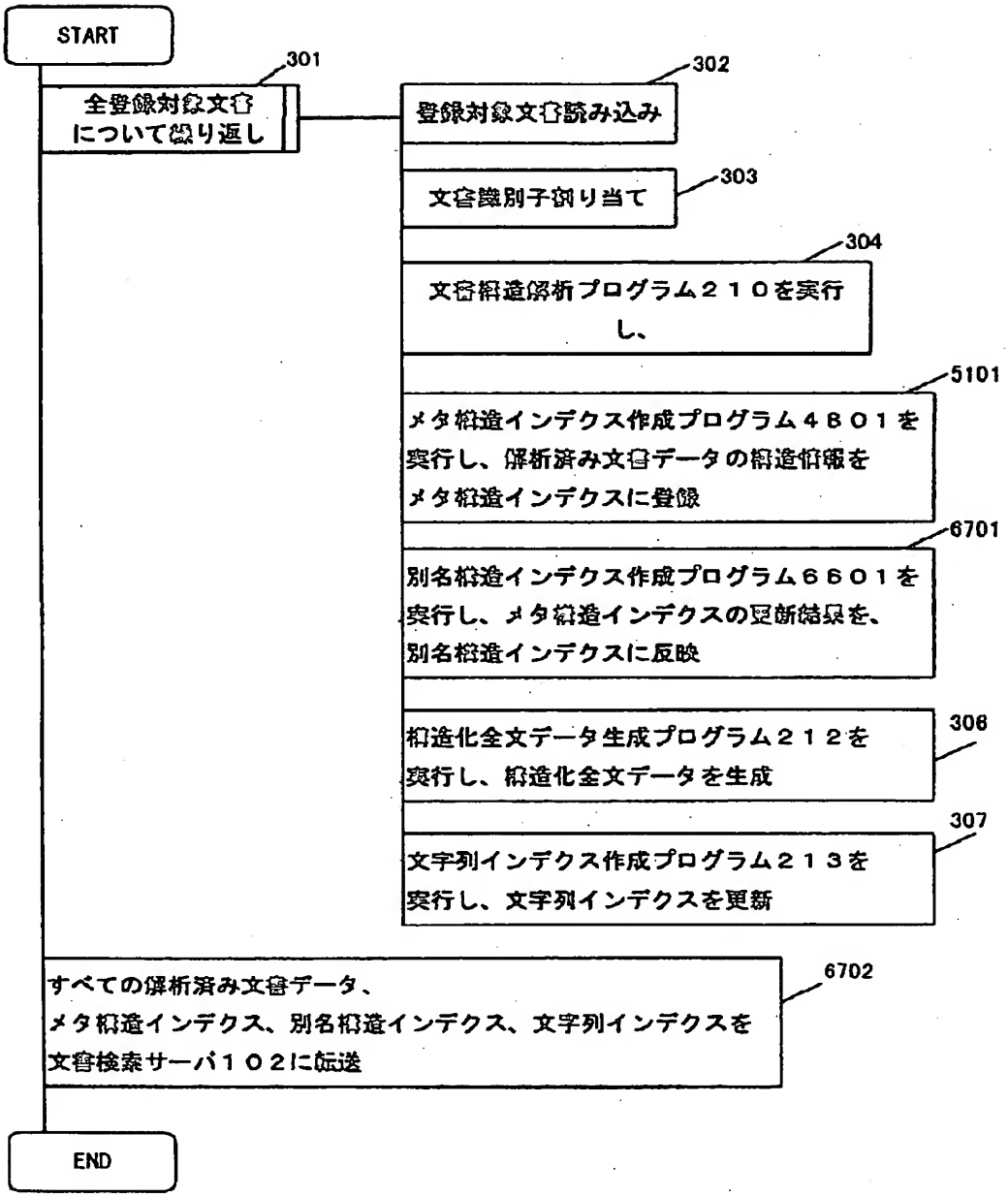
【図６６】

図６６


205
105
LAN/WAN
101
202
201
203
204
206
キーボード
ディスプレイ
CPU
FDD
通信制御装置
209
207
208
主メモリ
210
文書構造解析プログラム
4801
メタ構造インデクス作成プログラム
6601
別名構造インデクス作成プログラム
212
構造化全文データ作成プログラム
213
文字列インデクス作成プログラム
214
文書登録制御プログラム
215
システムプログラム
216
ワークエリア
解析済み文書データ格納領域
217
メタ構造インデクス格納領域
4802
構造化全文データ格納領域
219
文字列インデクス格納領域
220
組別定義テーブル
4803
別名構造インデクス格納領域
6602
別名定義テーブル
6603

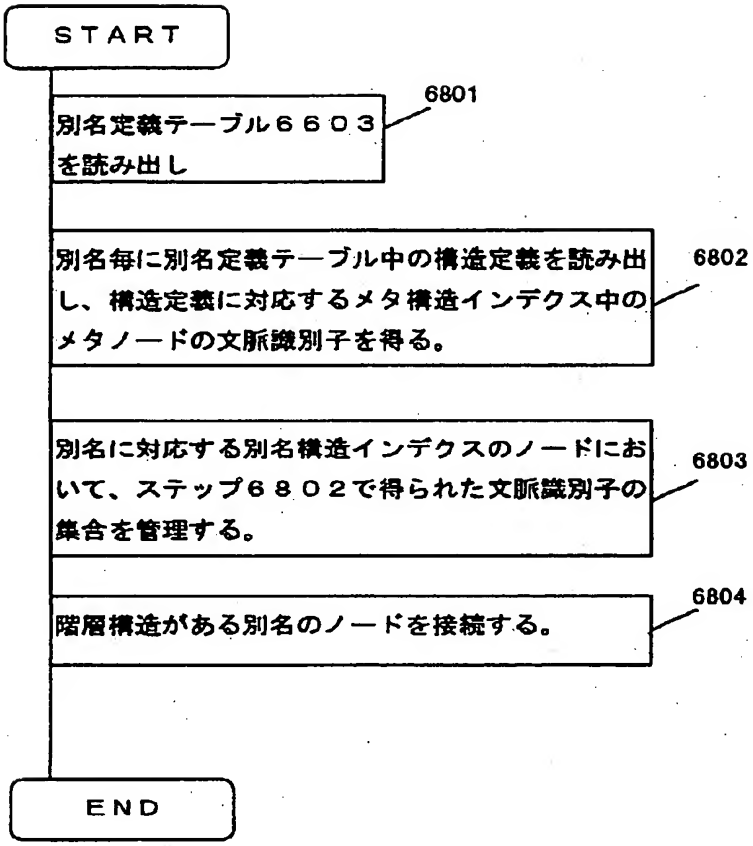
【図６７】

図６７


START
301
全登録対象文書について繰り返し
302
登録対象文書読み込み
303
文書識別子割り当て
304
文書構造解析プログラム２１０を実行し、
5101
メタ構造インデクス作成プログラム４８０１を実行し、解析済み文書データの構造情報をメタ構造インデクスに登録
6701
別名構造インデクス作成プログラム６６０１を実行し、メタ構造インデクスの更新結果を、別名構造インデクスに反映
306
構造化全文データ生成プログラム２１２を実行し、構造化全文データを生成
307
文字列インデクス作成プログラム２１３を実行し、文字列インデクスを更新
6702
すべての解析済み文書データ、メタ構造インデクス、別名構造インデクス、文字列インデクスを文書検索サーバ１０２に転送
END

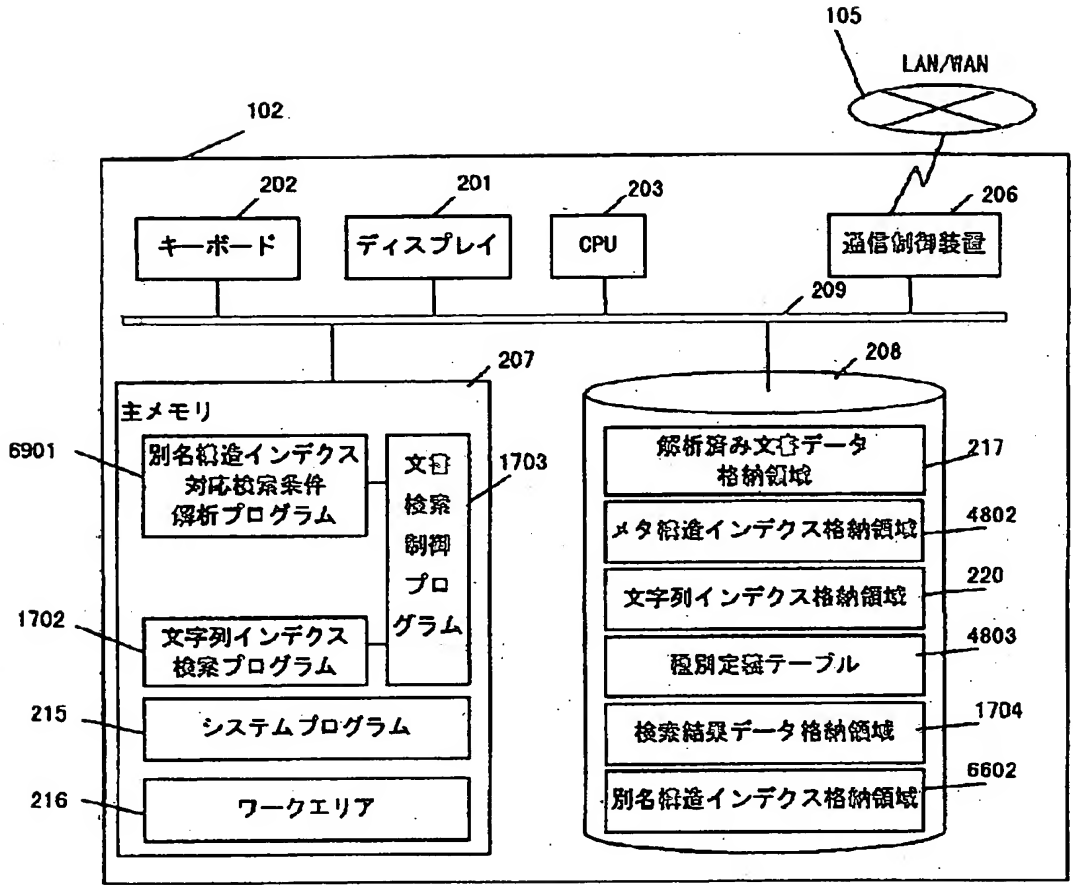
【図６８】

図６８


START
別名定義テーブル６６０３を読み出し
6801
別名毎に別名定義テーブル中の構造定義を読み出し、構造定義に対応するメタ構造インデクス中のメタノードの文脈識別子を得る。
6802
別名に対応する別名構造インデクスのノードにおいて、ステップ６８０２で得られた文脈識別子の集合を管理する。
6803
階層構造がある別名のノードを接続する。
6804
END

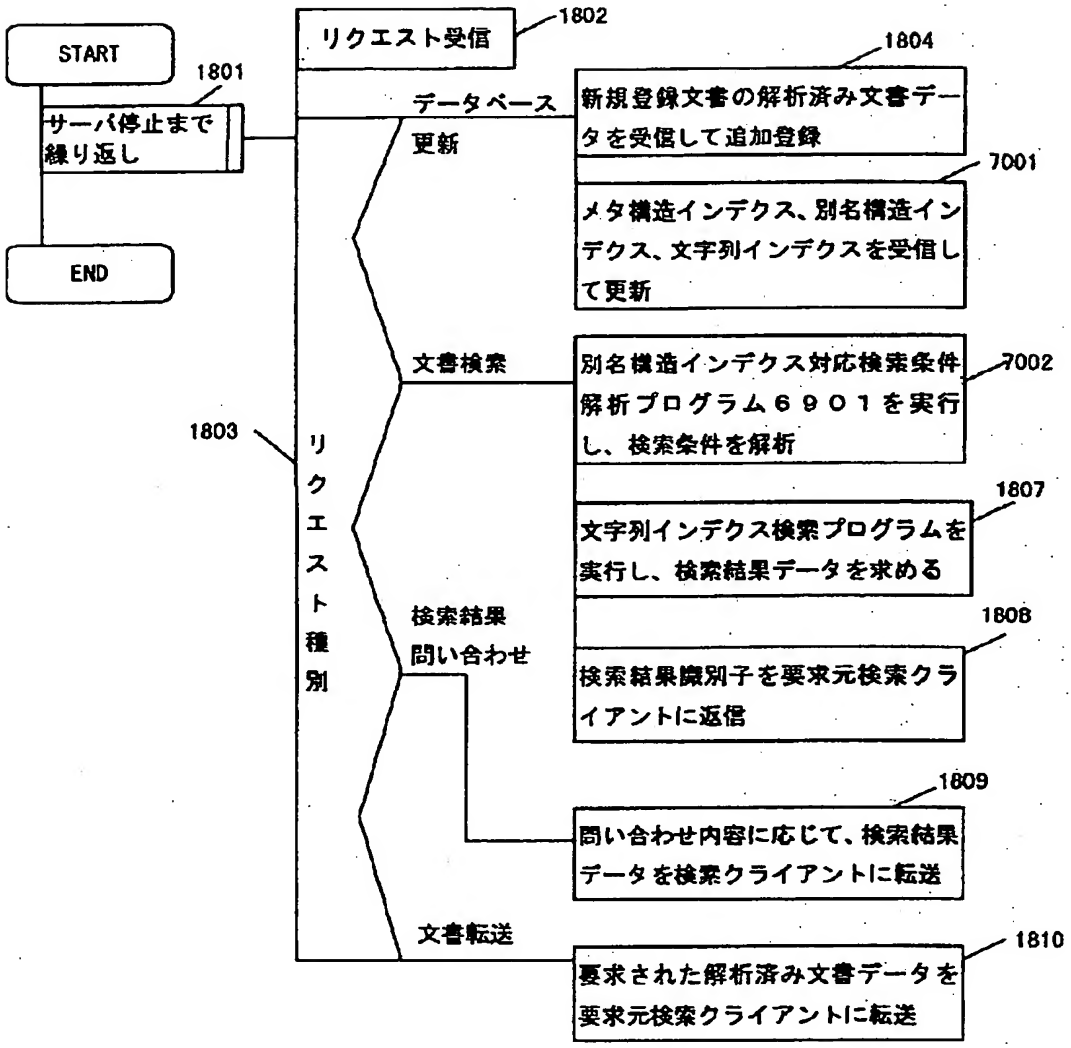
【図６９】

図６９


105
LAN/WAN
102
202
キーボード
201
ディスプレイ
203
CPU
206
通信制御装置
209
207
208
主メモリ
6901
別名構造インデクス対応検索条件解析プログラム
文書検索制御プログラム
1703
1702
文字列インデクス検索プログラム
215
システムプログラム
216
ワークエリア
解析済み文書データ格納領域
217
メタ構造インデクス格納領域
4802
文字列インデクス格納領域
220
種別定義テーブル
4803
検索結果データ格納領域
1704
別名構造インデクス格納領域
6602

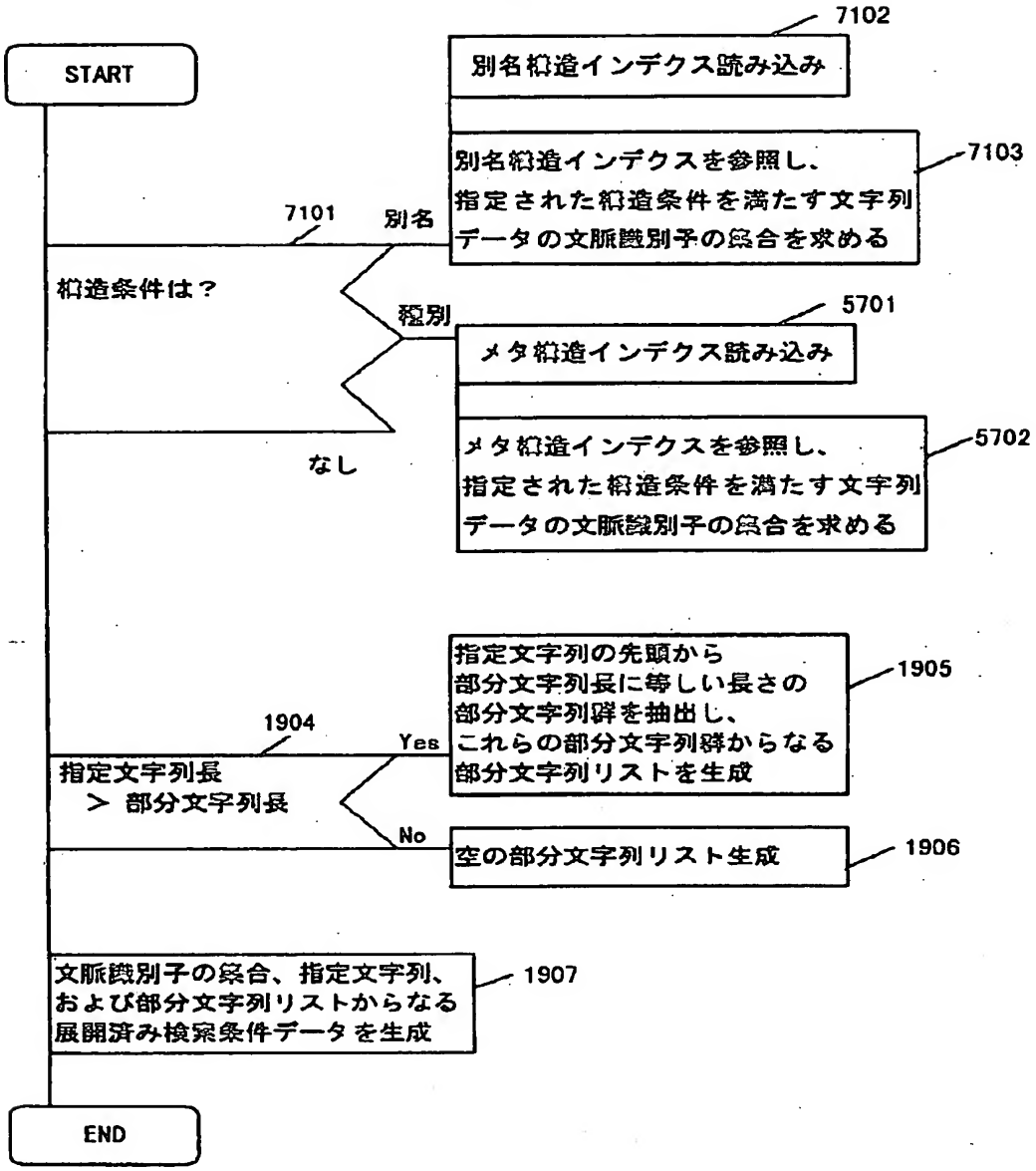
【図７０】

図７０


START
1801
サーバ停止まで繰り返し
END
リクエスト受信
1802
1803
リクエスト種別
データベース更新
文書検索
検索結果問い合わせ
文書転送
1804
新規登録文書の解析済み文書データを受信して追加登録
7001
メタ構造インデクス、別名構造インデクス、文字列インデクスを受信して更新
別名構造インデクス対応検索条件解析プログラム６９０１を実行し、検索条件を解析
7002
1807
文字列インデクス検索プログラムを実行し、検索結果データを求める
1808
検索結果識別子を要求元検索クライアントに返信
1809
問い合わせ内容に応じて、検索結果データを検索クライアントに転送
1810
要求された解析済み文書データを要求元検索クライアントに転送

【図７１】

図７１



フロントページの続き

(72)発明者　多田　勝己
神奈川県川崎市幸区鹿島田890番地　株式会社日立製作所情報通信開発本部内

(72)発明者　山崎　紀之
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内